Ecole Supérieure
d'Ingénieurs de Luminy
France

VICOMTech
San Sebastian
Spain

End of year internship

Development of new technologies for medical applications

Student : Charlotte Cottez
Referent teacher : Leon Mugwaneza
Company supervisor : Iñigo Barandiaran Martirena
Promotion : Computing 2007

Computing Department

IK4 Alliance

INIGraphics-Net

Acknowledgment

First of all, I would like to thank Julián Flórez and Jorge Posada for having received me in their company. This experience was really rewarding and it is thanks to them.

I would also like to thank Celine Paloc, who has always been kind and patient with me, as she was explaining me how the company was working. She was also very comprehensive for my doubts and helped me a lot when I had problems such as writing my report.

Then, I would like to thank Fernando Boto and Iñigo Barandiaran. I have worked with them during these 6 months, and everything has gone well with them. They were very patient and sweet with me. They taught me many things but also let me try on my own the ideas I could have. This was a real pleasure to work with them.

Finally, I would like to thank all the VICOMTech people for their kindness, patience (when I did not understand what they were explaining me) and their good mood. The ambiance was really cool and welcoming, it makes you feel staying.

Contents

List of Figures

Key words

Domain:

- Imagery
- Medical

Line of work:

- Applicative research

Technologies:

- Mathlab
- C
- Doxygen
- LaTeX
- Elliptical distance transform
- Random tree, random forest

Sort of work:

- Development
- Pattern recognition
- Classification

Overview

For the last year at the engineering school ESIL, in the computing area, a 6-months internship has to be made. As the image processing is a very interesting subject, and as an abroad experience is very rewarding, the internship takes place in a Spanish applied research centre called VICOMTech.

The work is the following : making research, developing the application, testing and interpreting the results.

This internship has two distinct subjects. The first one consists in detecting elliptical forms in an image. The other one is about a classification method called random forest. The first subject is used for a project called Zebrascreen, and the second one too, but it is also used for another project called Improve.

Zebrascreen is in collaboration with a biotechnological company. This project aims at developing an automatic screening of the position and the cardiac frequency of animal models. The prototype has to identify and classify microscopic images of fluorescence. The ellipse detection should have helped to identify the two chambers of the zebrafish's heart in a video. The resulting algorithm has a very high computational cost, and needs many parameters to work well. It will probably not be used for the identification of the chambers, however other applications are foreseen.

Improve is a European research project on Advanced Displays. The aim of this project is developing advanced technologies of visualization and interaction, on hardware and software levels. Developing markerless tracking and the corresponding classifier would allow to detect the coincidence between two images which come from a camera, in order to calculate its position and angle. The results are really good with generated examples, but with the real ones, it does not work very well. However, the problem is currently treated, and an article may be submitted about it at the end of September.

Chapter 1

Introduction

1.1 Context

For the last year at the engineering school ESIL, in the computing area, an internship of 6 months has to be made. As the imagery is a very interesting subject, I was looking for companies in this area. The thing is that I also wanted to go abroad, as this kind of experience is very rewarding. I could go to England, which could have been a good experience to improve my English, but the subject the working area were not as interesting as the proposed ones in VICOMTech. Moreover, VICOMTech was in Spain, near the frontier (so it could be possible to come back to France easily), and as it's been a long time that I had not spoken Spanish, it could be a good idea to improve my Spanish, so that I decided to make my internship at VICOMTech.

1.2 Description of the project

1.2.1 Initial topic

The project proposal at the beginning was to develop, during 6 months, functionalities in different applications. The different applications were on:

- Computer-Assisted Control of Endovascular Abdominal Aortic Aneuvrysm (AAA) Repair (EVAR). In this project, the work would have been to participate in the development of a software application to assist the medical team in the surveillance of EVAR patients.

- High-Throughput Content Screening Platform for use with Zebrafish (ZEBRASCREEN), the aim was participating in the development of a prototype used to identify and classify microscopic images of fluorescence.

- Collaborative Mixed Reality platform for Telepresence (IMPROVE): the work would have been to participate in the development of a collaborative mixed reality platform, more specifically on the software level (rendering, interfacing, transmission, visual tracking ect..)

1.2.2 Final topic

Finally, the work is on the two last projects and aims at developing two technologies. The first one is developing an algorithm which can detect the two chambers of the zebrafish's heart. The second one, which is used for both projects, is creating a classifier in order to, on the one hand, detect the objects which define an image coming from a camera, which allows to estimate its position and angle (IMPROVE project), and on the other hand, classify zebrasfish's phenotypes coming from a video (ZEBRAFISH project).

1.3 Company description

VICOMTech (VIsual COMmunication and Interaction TECHnologies) is an applied research centre for interactive computer graphics and multimedia located in the San Sebastian's Technological Park (Spain). It is a non-profit association, founded by INI-GraphicsNet. It is now incorporated into the IK4 alliance and is also classified as an Innovation and Technology Centre (ITC).

It is composed about 50 researchers and has an annual turnover of 2 million euros. It disposes about 60 applied investigation projects per year with local, national and European industry partners and administrations.

Its aim is to satisfy their clients' innovation needs. To do it, the Centre:

- works in applied research and in the development of multimedia technologies for visual interaction and communication.
- helps the mobility and formation of researchers.
- collaborates tightly with industries, universities and institutions, and also other technology centres.

Theire different application areas are:

- The Digital TV area, by exploring the possibilities offered by the synergies between Digital TV and Computer Graphics.
- Tourism, heritage and creativity. The aim of this area is to support the tourism industry and to promote both archaeological and heritage special interest sites, analyse and manage geographical information data.
- Interaction for edutainment and e-inclusion. This area focuses on the fields of education and leisure through the possibilities of audiovisual technologies and advanced user interfaces.
- Industrial applications. It brings the possibility to use advanced computer graphics technologies to different sectors of the industry and the society.
- The Biomedical Applications. It carries out research and development for the health care and biotechnology sectors.

The Biomedical's main research lines include the most recent advances in image analysis (image processing, segmentation, registration, fusion), visualization (virtual reality, augmented reality) and biomedical information management (transmission, representation, standards, interface).

(a) *Advanced video conference prototype* **(b)** *Vessels detection from a liver*

Figure 1.1. *Examples of biomedical applications*

VICOMTech benefits from the experience, the know-how and the support of the Fraunhofer-IGD and INI-GraphicsNet in the healthcare area. The unit has established close collaboration with local hospitals and private health clinics, and is working in strong synergy with a wide variety of clinicians (hepatic surgeons, pathologists, radiologist, radiotherapists). In the field of biotechnology, VICOMTech contributes to the local strategic initiative bioBASK 2010, working in different projects related to biosciences in collaboration with biotechnological companies and other research centres.

The objective of the department is to become a source of new technological and leading developments for healthcare services, and to satisfy the ICT needs of the biomedical community. In order to market its technology, eMedica S.L. has been founded in 2004 as a spin-off of VICOMTech.

The department can be divided into the 5 following working lines:

- Image Analysis: automatic image segmentation, multimodal image registration, measurement and characterization of biological processes.
- Advanced Interface: new visualization and interaction techniques like virtual reality and collaborative environment.
- Data Mining and Machine Learning: knowledge representation from large databases, image classification and retrieval, decision support system, biomedical knowledge discovery.
- Simulation: physically-based simulation, soft tissue modeling, interactions and surgical tasks, haptic interface,
- Telemedicine: tele-assistance and remote control, data transmission and security.

The project internship during these six months was working in collaboration with the Biomedical Application team on several projects, understanding and developing existing algorithms, and finding solutions. More specifically, the work was splitted in two different projects, ZEBRASCREEN and IMPROVE.

The ZEBRASCREEN project aims at developing a screening platform for evaluating drug effects and drug toxicity. This platform requires advanced image analysis methods for automated identification and classification of phenotypes. Such methods should allow to quantitatively measure complex phenotypes from thousands of images in a high throughput manner.

IMPROVE is an European research project addressing the strategic objective on Advanced Displays. Its aim is to develop advanced technologies of visualization and interaction, on hardware and software levels. Developing markerless tracking and the corresponding classifier would allow to detect the coincidence between two images which come from a camera, in order to calculate its position and angle.

Chapter 2

Ellipse detection

2.1 Description of the internship topic

2.1.1 Motivation

Nowadays, we can acquire many high resolution images of the organisms, but the current methods of analyzing these images is still mainly manual and visual. The lack of a computational approach to identify characteristics similarities is very costly for large-scale experiments and slows down the advancement in biology research, including genomic and drug discovery.

Consequently, automated systems for the interpretation of biological images would provide three important advantages over manual practice:

- high-throughput performance.
- quantitative and reproducible identification of phenotypes.
- consistent phenotypic information in databases.

Figure 2.1. *Example of a possible animal model*

Microscopic images very often present ellipsoidal structures, such as an animal head or cells. In the current project, the aim is to explore the use of advanced image analysis methods for automated identification and classification of such structures... Such methods should allow to quantitatively measure complex ellipsoidal structures from thousands of images in a high throughput manner.

2.1.2 State of the art

In the animal model case, several works have been made in order to identify the similarities of genetical expressions.

Just a few works have tried to identify and classify animal phenotypes automatically by using image processing. This results obviously that it depends on the structure type that it is pretended to be identified automatically, the difficulty level can vary considerably. For example, for the current project, in the ellipsoidal study, it would require the analysis of dynamical images (videos), which means that the complexity will increase inevitably. Consequently, accuracy and rapidity will be the main constraints of the algorithm.

2.1.3 Conceivable solutions

In the case of the image segmentation, additional problems such as the presence of noise and inevitable artefacts are present in an automatic acquisition system of animal models.

Finally, the aim is to segment dynamical and deformable structures, which can hide one behind the other one. In order to approach this problem, segmentation mechanisms based on image's textures, morphology or geometric are tested.

Organic structures such as eyes and heart chambers are ellipsoidal.. Consequently, the purpose will be investigate on algorithm which treat on elliptical structures with the same texture. In the cardiac chambers case, series of images will be used in order to complement the segmentation and in this form increase the algorithm robustness. This would also help to calculate the heart's cardiac frequency, which could help to detect heart problems (tachycardia or bradycardia), and so determine its phenotype.

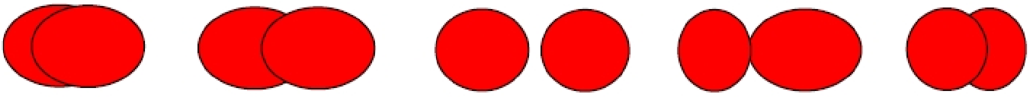


Figure 2.2. *Finding and extracting the 2 superposed elliptical structures*

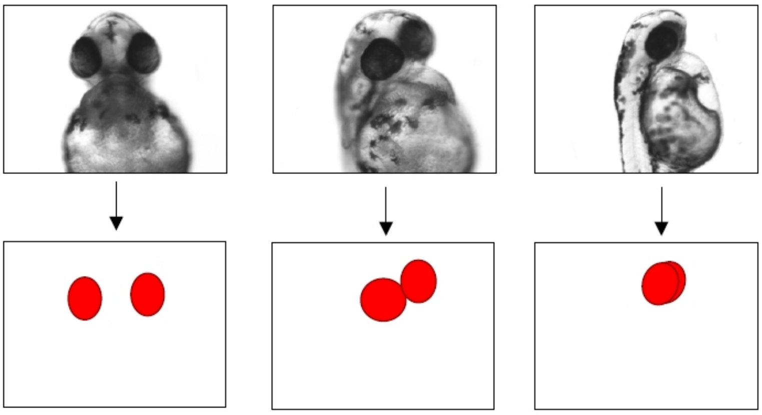


Figure 2.3. *Application on the eyes segmentation*

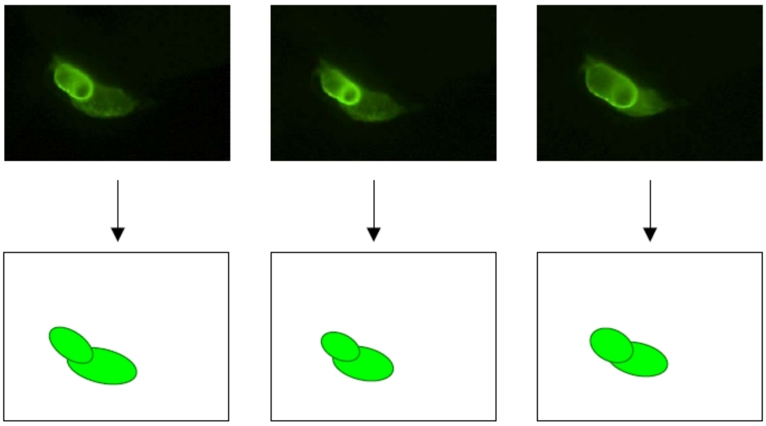


Figure 2.4. *Application on the cardiac chambers segmentation*

2.1.4 Proposed solution

The classical morphological method to separate fused objects in binary images is to use the watershed transform of the binary image. This method assumes roughly disk-like objects and cannot separate objects when they are fused together beyond a certain point. In the algorithm we want to evaluate [2], the issue is revisited by assuming that fused objects are unions of ellipses rather than mere disks. To do this, the pseudo-Euclidean distance transform algorithm is modified to generate arbitrary elliptical transforms in order to reduce the dimension of the problem and a goodness-of-fit measure is presented that allows to select ellipses. This algorithm is called the elliptical distance transform.

As the chambers of the heart and the animal's eyes have an elliptical form, we propose to evaluate its performance in our problem, as it proposes a solution of the separating fused binary objects problem.

2.2 Description of the implementation

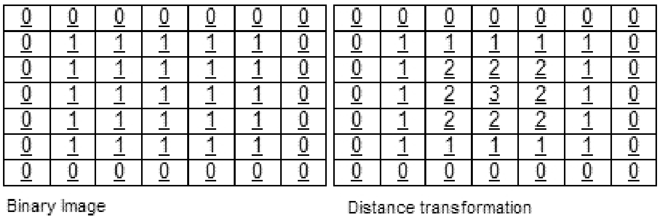
2.2.1 Technical description of the solution

As explained before, the selected algorithm is called elliptical distance transform. This method takes as a model many ellipses, with different parameters, and calculate for each of them a distance transform map, in order to find the centre of the ellipse. Then the most probable ellipse is taken. The selected algorithm is described as followed.

Elliptical distance map

First of all, an elliptical distance map, with given parameters (angle, grand axis, little axis) is created. What we call distance map is also known as distance transform or distance field. It is a digital image representation. The term choice depends on the point of view on the object we are looking at: whether the initial image is transformed into another representation, or it is simply endowed with an additional map or field.

The map supplies each pixel of the image with the distance to the nearest obstacle pixel. A most common type obstacle pixel is a boundary pixel in a binary image. See the image 2.5 for an example of a chessboard distance transform on a binary image.



Binary Image

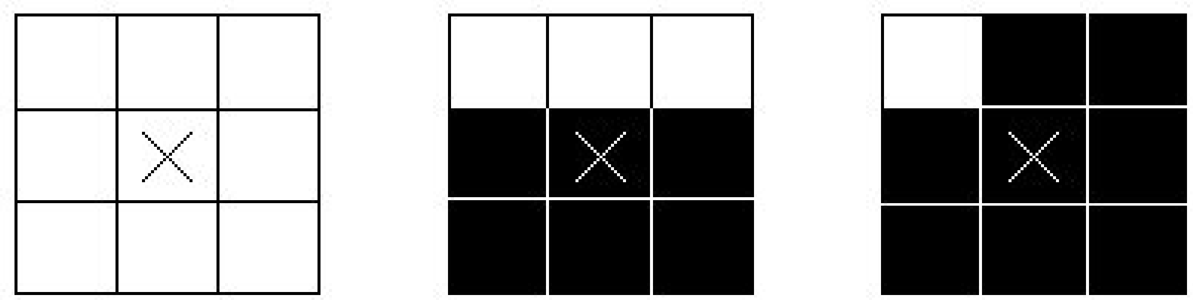
0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Distance transformation

0	0	0	0	0	0	0
0	1	1	1	1	1	0
0	1	2	2	2	1	0
0	1	2	3	2	1	0
0	1	2	2	2	1	0
0	1	1	1	1	1	0
0	0	0	0	0	0	0

Figure 2.5. *Chessboard distance transform*

In the selected algorithm, two vectors are used in order to know for each image point their distance from the background, according to their coordinates (X and Y). For example,



(a) *Background point. X(x, y) = 0 ; Y(x, y) = 0*

(b) *Limit point between the object and the background. X(x, y) = 0 ; Y(x, y) = -1*

(c) *Object point, but a little bit far from the background. X(x, y) = -1 ; Y(x, y) = -1*

Figure 2.6. *Possibilities for a pixel from an image*

In the Figure 2.6a, the nearest background point is himself, so that the distance from the background is 0 for the X and Y vectors.

For the Figure 2.6b, the distance from this point to the background is 0 for the X vector (as the nearest background point is right it up) and -1 for the Y vector (we look at the image from up to down).

For the Figure 2.6c, the distance from this point to the background is -1 for the X vector and -1 for the Y vector.

Such as for the chessboard distance calculus, the points which are connected to the background are first taken, and put into a queue with a priority value of 1 and we their distance from the background is calculated, in such a way as the method explained before. From now on, the same has to be done for all the uncalculated points.

To do this, the algorithm tells us that:

While the queue is not empty, the point with the most little priority value is taken (at first, all the points have a priority value of 1). This point will be called point C during the rest of the explanation.

If it is the first time that C is modified, the elliptical distance transform is calculated with its X and Y vector values (at this moment, they are filled of their distance to the background). For each C's neighbor, if it hasn't been modified before, and its X and Y values are defined, P1 is calculated with the elliptical distance with these X and Y values, otherwise, P1 is equal to infinity. P2 is also calculated as the elliptical distance between the C's distance to the background added to the distance value from C to its neighbor.

If P2 is less than P1, the neighbor is added in the queue with P2 as its priority value. The neighbor's X and Y values will be the C's X and Y values added to the X and Y relative difference between C and its neighbor. The neighbor's elliptical distance value is actualized to 1.

The elliptical distance formula is defined as following:

$$P(C) = \rho\sqrt{(b\cos(\theta - \alpha))^2 + (a\sin(\theta - \alpha))^2}$$

With the followed constraints:

- $\rho = \sqrt{X(C)^2 + Y(C)^2}$
- $Y(C) = \rho\sin(\theta)$
- $X(C) = \rho\cos(\theta)$
- $\theta = \arctan(Y(C), X(C))$

As you can see in the formula, different parameters, such as a, b and alpha are not calculated before. This is because they were given by the user before. These parameters correspond to the ellipse's parameters currently calculated (the ellipse's major axis, its minor axis, and its angle).

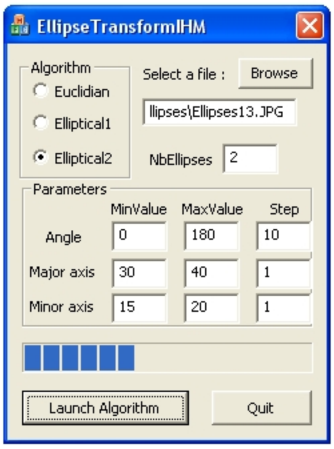
Finally, for all the points in the object, a value is given which corresponds to its elliptical distance from the background.

Finding ellipses

The highest values coming from the previous calculated elliptical distance map are taken. It corresponds to the most eligible center points for this ellipse (with its corresponding parameters). Consequently, all the points which define this ellipse are calculated and taken, and their euclidean distance value $P(C) = \sqrt{(X(C))^2 + (Y(C))^2}$ is calculated to determine how far they are from the background in order to know if the ellipse represents well the object or not (if the points are outside the object, their euclidean distance value will be the infinity). With all these values, we calculate the average value. It represents the general euclidean distance value for this ellipse. The most representative ellipse will be the one which has the most little euclidean distance.

2.2.2 Used methods and tools

In order to develop that algorithm, the program was first developed with MATLAB, because it provides many image processing functions. It has been developed in C++ with Microsoft Visual Studio 2005, in Windows XP, because MATLAB is too computationally expensive to execute the algorithm. The openCV library is also used, because, as MATLAB, it provides many image processing functions.



Algorithm
○ Euclidian
○ Elliptical1
● Elliptical2

Select a file : llipses\Ellipses13.JPG

NbEllipses 2

Parameters

	MinValue	MaxValue	Step
Angle	0	180	10
Major axis	30	40	1
Minor axis	15	20	1

Figure 2.7. *Ellipse detection HCI*

The image 2.7 represents the human-computer interaction (HCI) of the application. There are several parameters to enter before launching the computation.

- The selection of the algorithm. The Euclidean, the Elliptical or the normalized Elliptical distance transform.
- The image where we want to apply the algorithm.
- The number of ellipses the user wants to detect.
- From which angle to which angle the algorithm will be tested (and its step)
- From which major axis size to which major axis size the algorithm will be tested (and its step)
- From which minor axis size to which minor axis size the algorithm will be tested (and its step)

After having given all these parameters, the “Launch Algorithm” button allows to launch the calculus and give the selected image with its result.

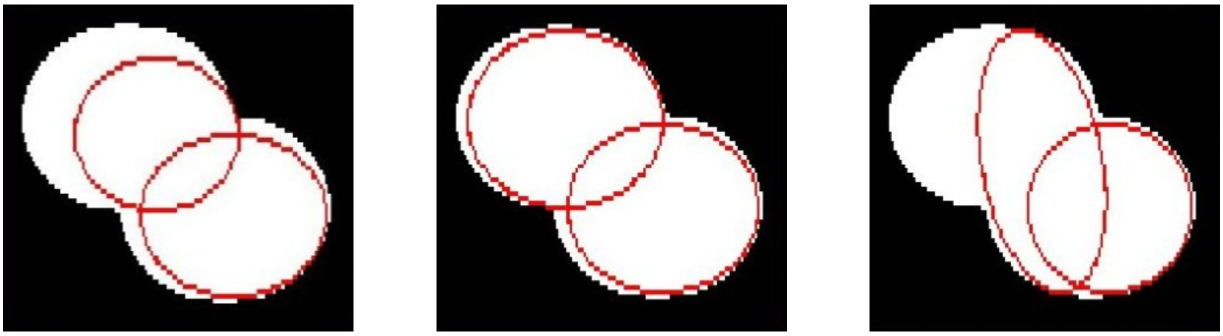
2.3 Obtained results

2.3.1 Global results

Two different ellipse transform methods were implemented: the elliptical distance transform, and its normalization. Moreover, for comparison purposes, a third method was tested: the Euclidean distance is not working. The corresponding formulas are the following:

- The elliptical distance transform: $P(C) = \rho\sqrt{(b\cos(\theta-\alpha))^2 + (a\sin(\theta-\alpha))^2}$
- The normalization of this formula: $P(C) = \rho\sqrt{\frac{\pi(cos^2(\theta-\alpha)+\sigma^2\sin^2(\theta-\alpha))}{\sigma}}$

- The euclidean distance: $P(C) = \sqrt{(X(C))^2 + (Y(C))^2}$

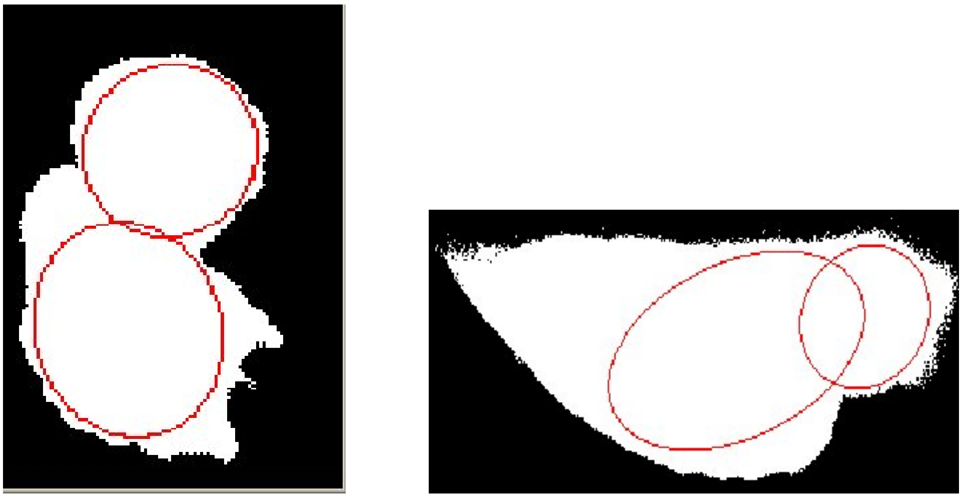


(a) *Elliptical distance* (b) *Normalized elliptical distance* (c) *Euclidean distance*

Figure 2.8. *Results with synthetic data*

The results of each method with synthetic data is showed at the Figure 2.8. We can see that the most accurate detection is the second one, with the normalized elliptical distance.

However, for the detection of the two chambers of the heart, we get a great error, as you can see it in the Figure 2.9.

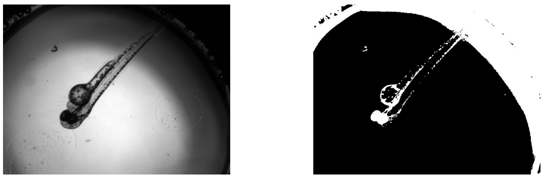


(a) *Experimental data: first heart image case. Angle: from 0 to 180, step 5 Grand axis: from 20 to 50, step 1 Little axis: until 10, step 1 Time: 3h16*

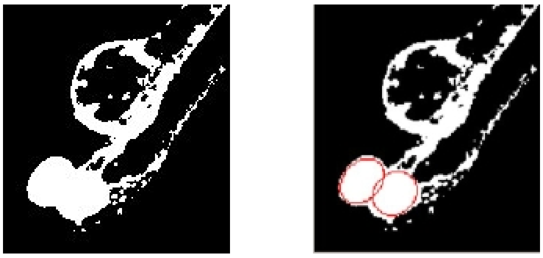
(b) *Experimental data: second heart image case. Angle: from 0 to 180, step 10 Grand axis: from 30 to 100, step 10 Little axis: until 15, step 10 Time: 40 min*

Figure 2.9. *Results with the images of the chambers of the heart*

However, it works better with more elliptical forms, as the zebrafish's eyes, as you can see in the figure 2.10d.



(a) *Initial image* **(b)** *Black & white image*



(c) *Cropped image* **(d)** *Result. Angle: from 0 to 180, step 5 Grand axis: from 5 to 20, step 1 Little axis: until 5, step 1 Time: 1h41*

Figure 2.10. *Results of the eyes' fish*

Actually, this method has a very high computational cost. For example, on the synthetic data, for detection two ellipses, with an angle from 0 to 180 with a 45 step, with a grand axis from 30 to 40 with a 1 step, and with a little axis from 15 to 20 with a 1 step, the execution time is 57 seconds, for a 129x170 size image. To detect the zebrafishs eyes, the algorithm takes 1h41 (for an image of 145x149 size).

2.3.2 Encountered difficulties

First of all, the algorithm is quite complex, so debugging was very long. There were several things that were not very clear, so we interpret it in the way that seemed to us the most appropriate one. For example, the article mentioned an angle θ but we did not know if it was in radians or in degrees. We decided to write to the author, M. Talbot, an associate professor at ESIEE in the A2SI lab since November 2004. His research activities are on image processing, optimization and algorithms. He helped us a lot, and this initiative has allowed to create a link between his university and our company. He sent us one of his student for an internship, and the company seemed to be very glad of him.

2.3.3 Eventual limits

As showed before, this method does not perform well with the heart's images, due to the poor quality of such image. Moreover, it was computationally very expensive and would take days in the case of image sequences. Finally,the method is not automatic, the number of ellipses and their size are required to launch the algorithm.

2.3.4 What the company can do with it

For the initial problem, the implementation cannot be used but the results are quite satisfying in other application, such as for example, identifying animal's eyes. This could help to know in which position the animal is. Moreover, the code could be optimized and used in other applications such as the segmentation of the abdominal aorte images or the identification of cells in microscopic images.

2.3.5 Suites and suggestions

The aim would be, in the future, to create an automated application, so that the computer can interpret himself the size and angle to test. Some improvement are also required regarding the execution time, which is too much long for the moment.

Chapter 3

Development of image classification methods based on computer vision

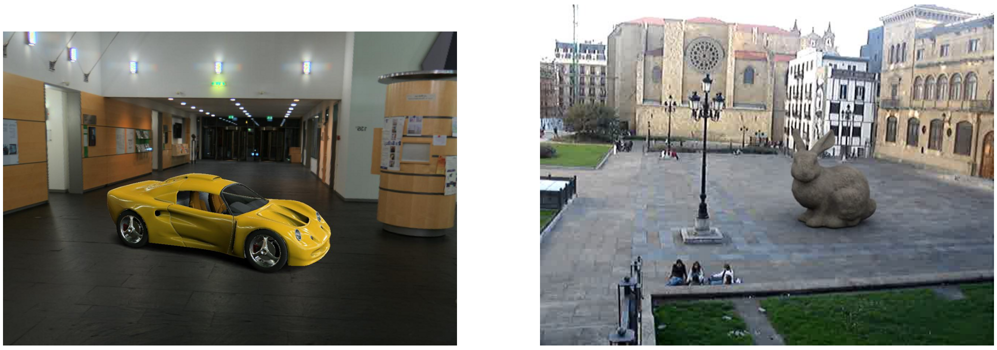
3.1 Description of the internship topic

3.1.1 Motivation

Zebrascreen

As explained before, the aim of this project is to determine automatically the observed animal phenotype. To do so, the local cardiac frequency needs to be computed, but also to be interpreted. Consequently, a classifier is needed to identify the phenotype by interpreting the local cardiac rate.

Improve



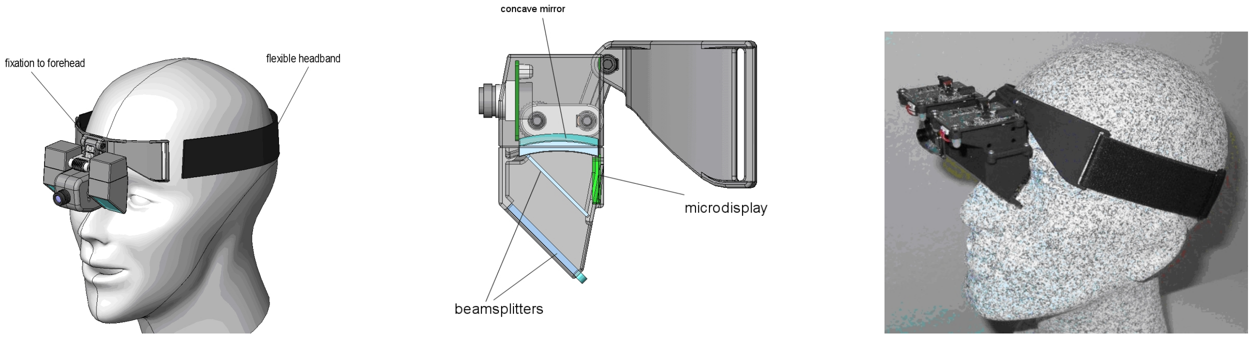
(a) *Indoor augmented reality* **(b)** *Outdoor augmented reality*

Figure 3.1. *Results: Creation of a mixed augmented reality*

The IMPROVE Project aims at developing advanced technologies of visualization and interaction, on hardware level (development of an HMD base on OLED), as well as software (improvements in rendering, 2D/3D interaction techniques for mixed reality, and advanced tracking system).

The achievements of IMPROVE are integrated into a collaborative mixed reality product development environment, showcased and evaluated in two application scenarios:

- collaborative product design in the car industry
- collaborative architectural design



(a) *HMD, fixation points to user's head* **(b)** *HMD, inside details* **(c)** *HMD, first prototype*

Figure 3.2. *HMD pictures*

The part of VICOMTech lays on the improvement of the estimation of the position and the orientation of a camera (related to a coordinate reference frame) integrated into a near-to-the-eye display, such as represented in Figure 3.2. It is supposed to create a mixed augmented reality, and work in a collaborative way (see Figure 3.3).

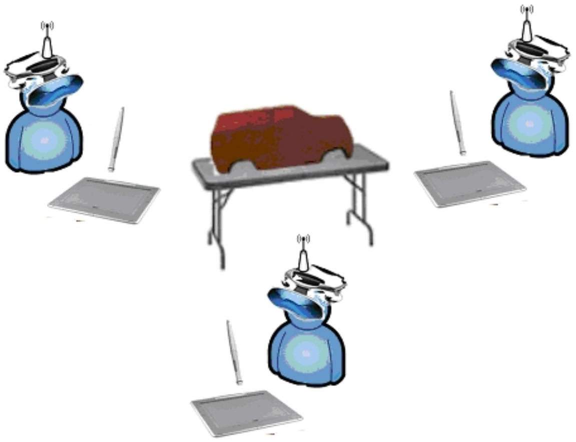


Figure 3.3. *Collaborative mixed augmented reality*

A camera tracking based on images is a method that allows to estimate the position and orientation of a camera related to a coordinate reference frame, by analyzing an image sequence.

In particular, the estimation of the camera pose is obtained by matching or recognizing a set of points extracted from the images, taken from the same object or environment (3.4).

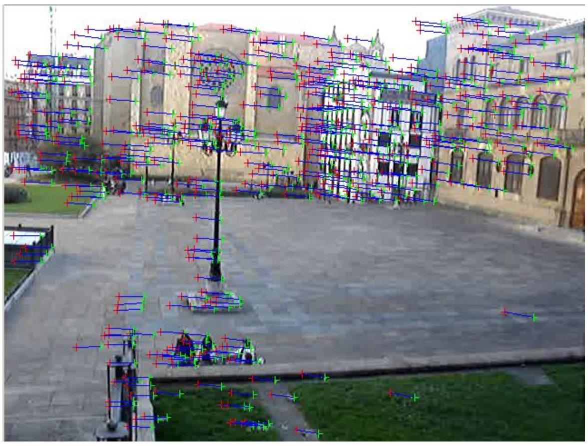
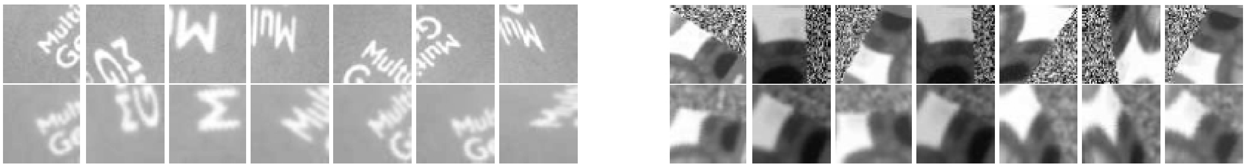


Figure 3.4. *Markerless tracking*

During the estimation process, some points, features or descriptors must be extracted from each image. Once these points are obtained, they must be matched with those points learnt during the training phase (tracking by detection) or with those extracted in the previous image (recursive tracking). Both tracking by detection and recursive tracking solve the problem of the estimation of the camera transformation (pose) but from two different approaches.



(a) *Patches centered at a keypoint extracted in several new views, synthesized using random affine transformation and white noise addition*

(b) *Patches after orientation correction and Gaussian smoothing*

Figure 3.5. *Patches with different applied geometrical transforms, used in the training phase*

In this project, we are interested in the tracking by detection approach. In this type of methods, a previous training phase, before the camera pose estimation process takes place, is needed. In this training period, the system is trained with some known images of the objects or the environment where the camera will be during pose estimation process. These images are modified with geometrical transformations, in order to have new views from them (3.5). The knowledge of the environment increases the overall robustness of the system, and give the results showed at 3.6, and allow to create a mixed augmented reality, in indoor (3.1a) and outdoor environments (3.1b).

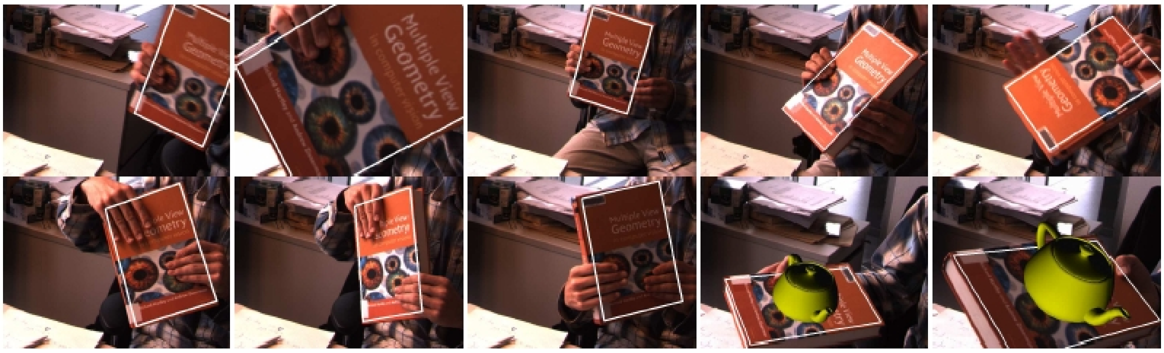


Figure 3.6. *Detection of a book in a video sequence*

Conclusion

For these two projects, we have to automatically classify:

- Functional phenotypes, using different characteristics such as the two local cardiac frequencies.
- Recognition keypoints, which have also many characteristics, such as the texture, the form, and so on.

These classification will help to make correspondences with :

- The characteristics which define each phenotype
- Other recognition keypoints we had defined before.

3.1.2 State of the art: Classifiers

Generality

Machine learning techniques can be splitted in two different groups, known as Supervised Classification and Unsupervised Classification.

In both cases, the problem is to find the observed data's class (also called "special feature"). Each class that forms part of the problem is composed by several independents characteristics (also called "features" or "variables"). The data is also often called "descriptor". These variables define the class, so that the special feature can be found thanks to them, as they represent the characteristics of the class.

Let's explain it with an example: let's say that the given descriptors are a group of people. The classification problem is detecting healthy people and ill ones. In this example, the special feature (or class) is the health of the patient, which may have two different values, healthy or ill, and the independent variables could be the symptoms the patients may present such as, headache, cold,etc.

Unsupervised classification

The main difference between supervised and unsupervised classification is that unsupervised learning is based on observations and interpretations, whereas the supervised learning needs model data (they are given during what it is called the training period). In unsupervised classification, the program has to find by himself the data structure, with the class and its corresponding features. Every input objects are treated as a set of random variables, and then regrouped in classes according to their similarities.

The most common unsupervised classifiers are :

- Neural networks (but not all of them)
- Clustering

Supervised classification

During a supervised classification process (papers [4] ans [3]), the independent variables of every object which have to be classified are analyzed, and the decision to which class (among the possible ones that had been defined before), the object belongs to is taken. This type of techniques requires a previous process, in which a system is trained with a determined set of known examples (training set) that present variations in all their independent variables. Once the process is finished, the system is trained and ready to classify new examples.

It exists many available classifiers. Each one has its strengths and weaknesses. Classifier performance depends greatly on the characteristics of the data to be classified. There is no single classifier that works best on all given problems, and determining a suitable classifier for a given problem is still complicated.

The most widely used supervised classifiers are the following ones :

- The Neural Network (Multi-layer Perceptron)
- Support Vector Machines
- k-Nearest Neighbors
- RBF classifiers
- Decision Tree

3.1.3 Conceivable solutions: Decision tree

For the two projects, the work consists in the development of supervised classification techniques. More specifically, the study is on a classification technique known as “classification tree” or “decision trees”.

This type of classifiers are known as tree because of their hierarchy structure. This structure is built up starting from a special point, known as the root node, and the descending nodes, known as children nodes, such as the figure 3.7.

When the tree is constructed, at the beginning, it disposes of all the training data, with their independent variables and their corresponding class. The aim of the tree is that, at the end,

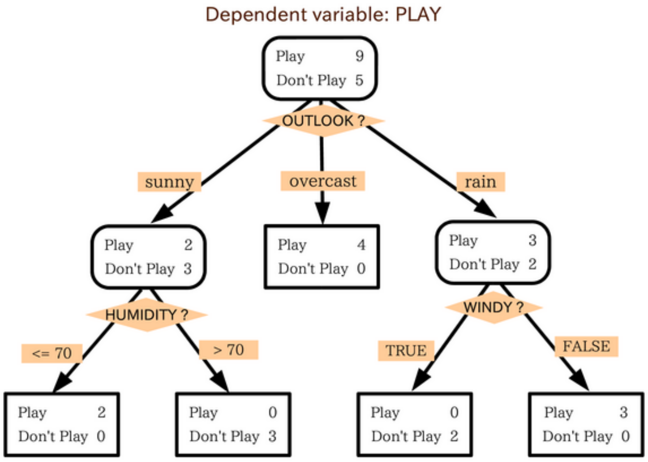


Figure 3.7. *An example of decision tree: people, according to the weather, play or don't play*

it can classify correctly a given data. In order to do it, on every tree's node is a discriminant criteria which allows to know to which child the descriptor has to go. This criteria belongs to one of the independent variables, or to a set of them. This set is chosen according to a value called the information gain. In machine learning this value can be used to define a preferred sequence of attributes to investigate rapidly the state of a given data. Usually an attribute with high information gain should be preferred to other attributes.

3.1.4 Selected solution: Random forest

In the Zebrascreen project, we are currently developing a prototype aiming at identifying and classifying cardiac phenotypes of the zebrafish. This type of techniques requires a previous process, in which a system is trained with a determined set of known cases (training set) that present variations in all their independent variables. Once the process is finished, the system is trained and ready to classify new examples.

In the Improve project, the aim is to detect in real-time the different objects which composes the images given by the camera (papers [7] and [6]). To do so, the system is trained in a preprocessing step. Assuming that several registered target object's images are available, a keypoint-based approach has been developed in order to match keypoints extracted from the input images to those found in the model images. This is a classification problem. Consequently, the problem is to find a robust, accurate, and fast enough classifier for frame-rate performance.

The applied algorithm is called "random forest" [5]. They naturally handle multi-class problems and are robust and fast, while remaining reasonable easy to train. Its particularity is that it is a multi-classifier, as it is composed of many decision trees (and so that many classifiers) and outputs the class that is the most selected in individual trees.

This type of structure is known as random because in each node of the tree an exhaustive test of the independent variables is not carried out, but a random selection of them. Moreover, the selection of training images in every tree is taken randomly. When the number of different classes and the number of independent variables in each class is high, an exhaustive analysis is not feasible. That is the same for the number of images. Therefore, some branches of the tree must be deleted. The random tree approach reduce this classification cost by doing a set of random selections of the independent variables to be analyzed or tested in each node.

There is a difference between the standard random forest and our program. Instead of selecting randomly the set of independent variables, a set of several random variable sets is made, and, thanks to the information gain value, the set which reduces the most the tree complexity is taken.

The advantages of random forest are:

- For many data sets, it produces a highly accurate classifier.
- It handles a very large number of input variables.
- It estimates the importance of variables in determining classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It includes a good method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It provides an experimental way to detect variable interactions.
- It can balance error in class population unbalanced data sets.
- It computes proximities between cases, useful for clustering, detecting outliers, and (by scaling) visualizing the data.
- Using the above, it can be extended to unlabeled data, leading to unsupervised clustering, outliers detection and data views.
- Learning is fast.

3.2 Description of the implementation

3.2.1 Technical description of the solution

Construction and tests

The first part consists in the construction of the forest. To do it, the training and the test sets have to be determined. Let's say that there are Tr training data and Te tests descriptors.

Each descriptor is composed of M independent variables and a special feature. During the construction of the tree, each node is treated as following:

1. tr training descriptors are in this node.
2. S random sets of m descriptor's variables are selected. (m is given by the user and is less than M)
3. For each set, its information gain is calculated.
4. The variables set with the greatest information gain value is selected (s).

5. For each descriptor, the values of the s' variables are added and gives a new value.
6. The greatest possible number of children are created and share, in an equally manner, all the descriptors, by compare at their value.
7. The same is done for the children nodes, until the sharing is not possible anymore (there is only one represented class, there is only one value, or the node is at its maximal depth). At this moment, thanks to the remained descriptors, the representation probability is calculated for each class.

An example of a tree construction is given at 3.8 and the main steps of its construction are at 3.9.

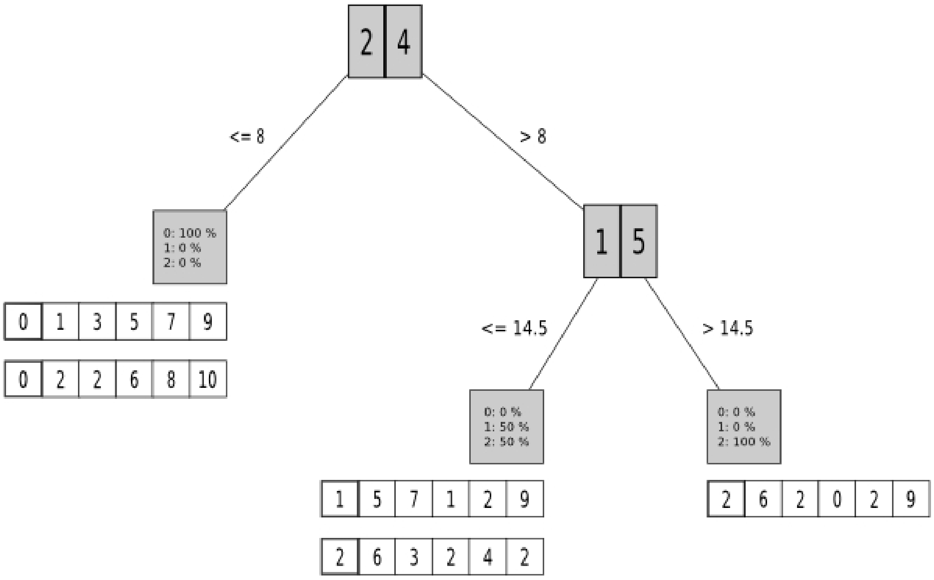
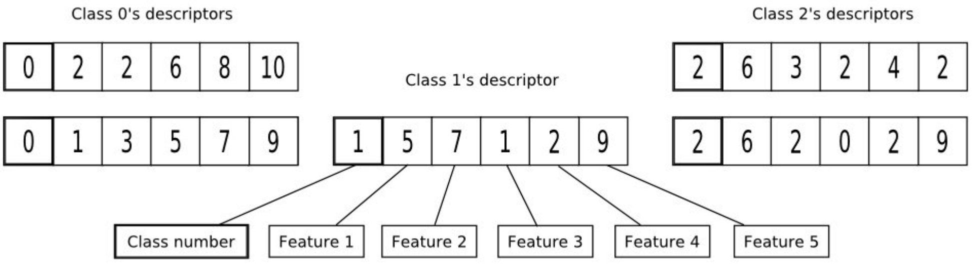
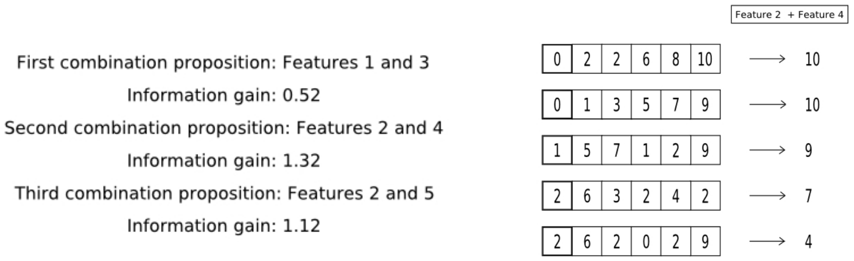


Figure 3.8. *An example of a tree construction*



(a) *These are the descriptors input*


First combination proposition: Features 1 and 3

Information gain: 0.52

Second combination proposition: Features 2 and 4

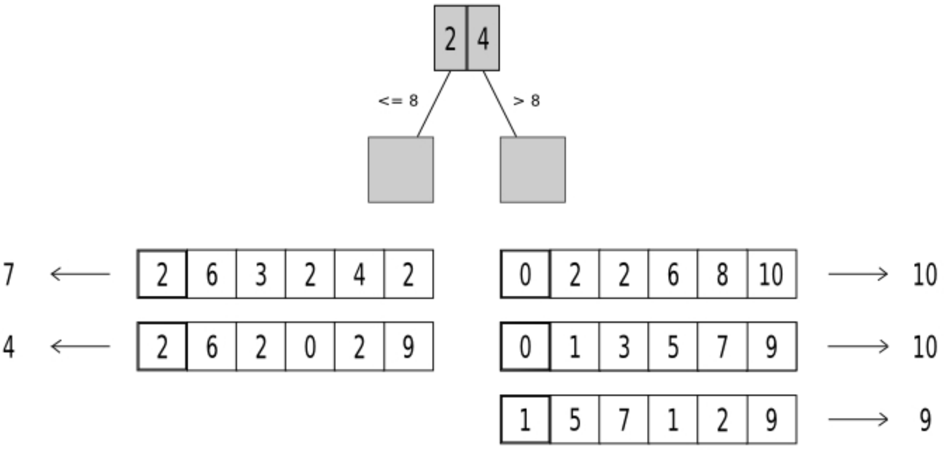
Information gain: 1.32

Third combination proposition: Features 2 and 5

Information gain: 1.12

(b) *3 sets of combination of two features are taken randomly. The resulting information gain is calculated*

(c) *The combination with the greatest information gain value is taken and the corresponding values are calculated for each descriptor. Here, we can see that we have 4 different values. As the maximal children number is 2, there will be only one split, between 7 and 9*



(d) *Construction of the children, and repartition of the descriptor in their corresponding child*

Figure 3.9. *Main steps in the tree construction. Here, the maximal number of children is 2 and the maximal depth is 3. The number of features used to create the combination is 2.*

Information gain calculus

The order in which attributes (or attributes combinations) are chosen determines how complicated the tree is. Consequently, the most informative set of attributes has to be taken. The information content of a data has to be related to the probability in receiving this type of data. Data with a high probability of arrival are not as informative as messages with low probability. Learning aims to predict accurately, which means reduce the low probabilities. An existing calculus allows to give a value for every attribute (or combination of attributes) which corresponds to his degree of information. It is called the “information gain” and the attribute (or the combination of it) which has the greatest information gain will be the most informative attribute (or attributes combination) there can be. This information gain is calculated as following [8]:

Assume there are k classes $C_1, ...C_k$. The aim is to decide in which set of attributes is the best to split on.

The information gain from splitting N data on variable A is calculated in such a way:

- The entropy (E) of the current set of data is calculated .
- For each value a_j of the variable $A_{(j=1,...,r)}$
  - There are
    - $J_{j,1}$ instances in class C_1,
    - ...,
    - $J_{j,k}$ instances in class C_k,

    for a total of J_j instances with $A = a_j$.
  - Let
    - $q_{j,1} = J_{j,1}/J_j$,
    - ...,
    - $q_{j,k} = J_{j,k}/J_j$
  - The entropy E_j associated with $A = a_j$ is $E_j = q_{j,1} \cdot log_2(q_{j,1})... - q_{j,k} \cdot log_2(q_{j,k})$
- Finally, the information gain associated with a split on attribute A $IG(A) = E - (J_1/N) \cdot E_1...(J_r/N) \cdot E_r$.

The entropy E of the current set of instances is calculated in such a way:

- There are N data classified to this node
  - I_1 belong to class C_1,
  - ...,
  - I_k belong to class C_k,
- Let
  - $p_1 = I_1/N$,
  - ...,

- $p_k = I_k/N$,
- Then the initial entropy is $E = -p_1 \cdot log_2(p_1)..-p_k \cdot log_2(p_k)$.

Let's explain it with an example. There are 7 instances, described in three features or independent variables (size, color, and shape), and data are classified into two classes, yes and no.

For the “yes” class, we have the following descriptors :

- medium blue brick
- small red sphere
- large green pillar
- large green sphere

For the ”no” class, we have the following data :

- small red wedge
- large red wedge
- large red pillar

Global number of positive data: I_1	4	Global number of negative data: I_2	3
Global positive probability $p_1 = I_1/N$	0.57	Global negative probability $p_2 = I_2/N$	0.43
Total number of data: N	7	Total number of classes: C	2
Global entropy $E = -\sum_{i=1}^{C} p_i \cdot log_2(p_i)$		0.99	

Table 3.1. *Global data*

	SIZE (F = 3)			**SHAPE (F = 4)**				**COLOUR (F = 3)**		
	Small	Medium	Large	Brick	Wedge	Sphere	Pillar	Red	Blue	Green
Number of data: J_j	2	1	4	1	2	2	2	4	1	2
Number of positive data: $J_{j,1}$	1	1	2	1	0	2	1	1	1	2
Number of negative data: $J_{j,2}$	1	0	2	0	2	0	1	3	0	0
Positive probability: $q_{j,1} = \frac{J_{j,1}}{J_j}$	0.5	1	0.5	1	0	1	0.5	0.25	1	1
Negative probability: $q_{j,2} = \frac{J_{j,2}}{J_j}$	0.5	0	0.5	0	1	0	0.5	0.75	0	0
Entropy $E_j = -\sum_{i=1}^{F} q_{j,i} \cdot log_2(q_{j,i})$	1	0	1	0	0	0	1	0.81	0	0
Information gain $IG(A) = E - \sum_{i=1}^{F} E_i \cdot (J_i/N)$	0.13			0.7				0.52		

Table 3.2. *Information gain calculus*

Consequently, as it can be seen in the tables reftable, the best information gain is on the shape feature, so that the split will be on this variable. Then, the operation is repeated for all subtree, and the result is showed on the figure 3.10.

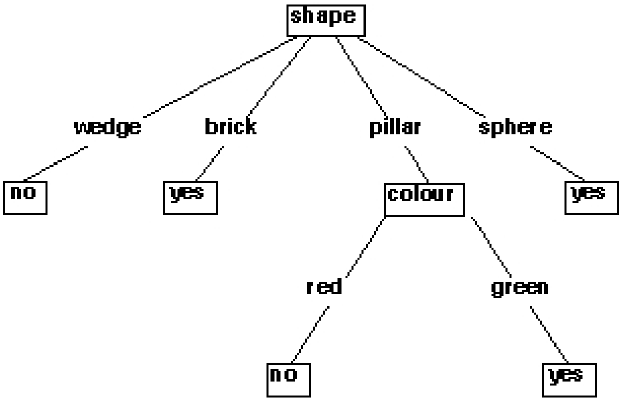


Figure 3.10. *Result by using the gain information splitting*

Test period

After having constructed the trees, there is the test phase, the moment where the accuracy of the tree is tested. In fact, P test descriptors are given in input, with their features but also their corresponding class, their label. The forest's aim is, thanks to the features, to find the corresponding class. To do it, in each tree, the tested descriptor is given in the root node. As seen before, during the tree construction, the used attributes combination is given. Moreover, in each child is given the maximum and the minimum values in which the calculated value (the addition of attributes' values - these attributes correspond to the attributes combination given in the father's node-) has to be so that the descriptor belongs to this child. Consequently, until a leaf is not encountered, the tested descriptor goes down in the tree, according to the child it belongs to. At the end, a leaf is encountered, which contains, for each class

- The number of data which are in this node.
- For each class, the number of data which correspond to this class
- For each class, its probability (obtained thanks to the two previous informations)

Consequently, the given probabilities (and number of occurrences) have to be interpreted, in order to have the most accurate result, but many selection methods exist, and give different results. So that selection method tests have to be made, in order to find the most accurate selection method for this classifier.

10 methods have been developed:

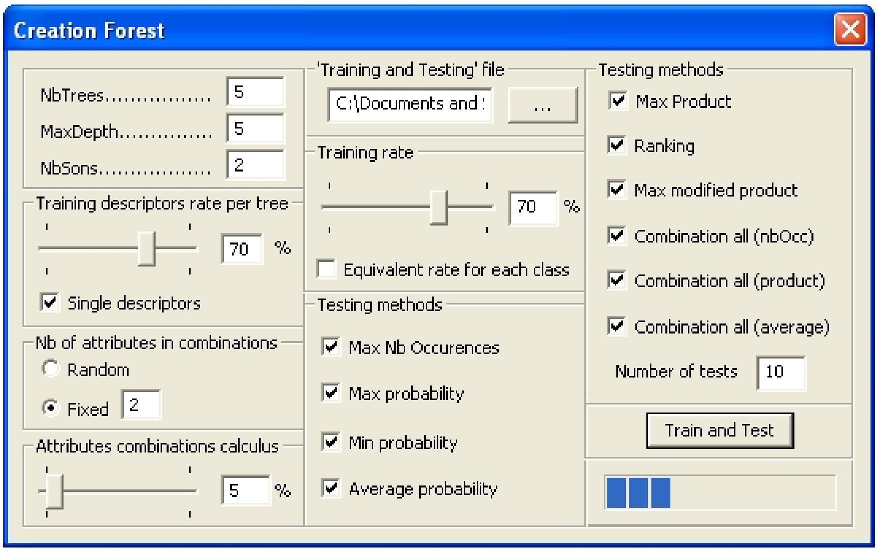
1. The maximal number of occurrences method: it consists in taking in every tree the greatest number of occurrences (and its corresponding class), add one to the corresponding class (as it corresponds to one occurrence in the whole forest), and then taking the greatest number of occurrences (and its corresponding class) among all.

2. The maximal value of maximal values of probabilities method: in every tree, the greatest probability and its corresponding class are taken. The most probable class will correspond to the one who has the greatest probability among all the trees.
3. The maximum value of minimum values of probabilities method: in every tree, the most little probability and its corresponding class are taken. The most probable class will correspond to the one who has the greatest probability among all the trees.
4. The maximal value of average values of probabilities method: it consists in calculating the probability average of each class, and then taking the greatest value.
5. The maximal value of probabilities product method [12]: in this case, all the probabilities of each class are multiplied, and the greatest value is taken.
6. The maximal value of probabilities modified product method: This method is quite the same as the previous one, instead of one supplementary constraint [11]:
   (a) For each class, we find the number z of experts the output of which is below a defined threshold t.
   (b) If z is less than half the number of experts R, then the experts outputs that are below the threshold value are modified by setting their output to the threshold.
   (c) If z = 0 or is larger than half the number of experts, then the outputs experts remain unaltered.
7. The ranking method [10]: it consists in putting a trust weight on the probabilities. This will depends on the value of the probability. In fact, in each leaf, the greatest probability will have the greatest trust weight and the most little one will have the most little value. After this, every class probability (multiplied by its trust weight) is added. Finally, we take the greatest value, and its corresponding class.
8. The combination of all the previous methods : Maximal number of occurrences: in this case, we use all the previous methods and their results (the selected class), and the number of times this class has been chosen is calculated. Then the most eligible class will be the one which was the most selected.
9. The average of all the previous methods : Maximal probability: here, all the probability calculated with all the previous method are taken, and are used to calculate the average of these probabilities. Then, the class which has the greatest value is selected.
10. The product of all the previous methods : Maximal probability: here, all the probability calculated with all the previous method are taken, and are multiplied. Then, the class which has the greatest value is selected.

3.2.2 Used methods and tools

The program is developed in C++ with Microsoft Visual Studio 2003 and commented with Doxygen. It has an MFC interface, in order to create the forest and to test it, with different parameters. The generation of fictive examples was developed in MATLAB. The training and test files are in format .txt. They are composed of :

- Number of classes
- Number of features for each descriptor
- Number of descriptors in the file
- All the descriptors with :
    - The number of the corresponding class
    - The features with their values



NbTrees................ 5
MaxDepth.............. 5
NbSons................. 2
Training descriptors rate per tree 70 %
☑ Single descriptors
Nb of attributes in combinations
○ Random
● Fixed 2
Attributes combinations calculus 5 %
'Training and Testing' file C:\Documents and :
Training rate 70 %
☐ Equivalent rate for each class
Testing methods
☑ Max Nb Occurences
☑ Max probability
☑ Min probability
☑ Average probability
Testing methods
☑ Max Product
☑ Ranking
☑ Max modified product
☑ Combination all (nbOcc)
☑ Combination all (product)
☑ Combination all (average)
Number of tests 10

Figure 3.11. *Classification HCI*

The Figure 3.11 represents the classification humancomputer interaction (HCI). It is composed of many parameters needed to construct the forest that can be changed, such as

- the number of trees in the forest
- the maximal number of children for a node
- the maximal depth
- the rate of training descriptors used for a tree construction
- having or not single training descriptors in each tree
- the number of attributes used in attributes combination. This number can be taken randomly every time.

- the number of combination possibilities which are going to be calculated, in each construction node. This value is a rate of all the combination possibilities which can be calculated.

There is also the selection of the training and test file which contains all the descriptors and their repartition between the training period and the test period can be modified by using the "training rate" bar.

Finally, there is also a test part, with the different selection methods implemented. They can also be selected or not. When the button "Create and test" is pressed, the forest construction is made thanks to the training descriptors and then tested with the test descriptors. The forest construction and tests can be done many times, thanks to the number in "Number of tests". The results are sent in a new file called "filename STAT.txt"

3.3 Obtained results

3.3.1 Global ones

Synthetic data set

Many different parameters have been tested to see which ones could influence the most the results. Consequently, we have modified the number of trees for example, but also the number of children and the number of attributes selected and the percentage of possibilities we wanted to calculate and finally the selection method. The results have been compared with the Weka software and they are quite the same but for a lower execution time. This has been tested on different files, with different number of classes, attributes, attributes values and descriptors, but we will only one file to present the obtained results:

- There are 30 classes
- There are 5 features
- There are about 33 descriptors for each class

First of all, we try different selection methods to see if it could influence the results. The default parameters are:

- The number of random trees is 10
- The number of children is 3
- The maximal depth is 10
- In every node, 2 attributes are selected for the calculus
- We only calculate 90 % of all the possibilities
- There are 70 % of descriptors for the training period and the rest for the test period

Maximal number of occurrences	Maximal probability	Maximum of the minimal probability	Maximum of the average probability	Maximal product of the probabilities	Maximal modified product
98.33	56.33	19.33	99.00	97.67	97.67
99.00	61.67	18.00	99.33	97.67	97.67
100.00	61.67	16.33	100.00	99.33	99.33
97.67	57.67	16.33	97.33	98.67	98.67
99.67	59.00	19.67	99.67	99.33	99.33
98.67	57.00	18.00	99.00	98.67	98.67
98.00	55.67	18.33	98.33	97.00	97.00
98.33	59.33	19.67	98.67	96.67	96.67
98.33	65.00	21.00	98.33	97.67	97.67
99.33	59.00	18.67	99.33	98.33	98.33
98.73	59.23	18.53	98.89	98.10	98.10

Table 3.3. *Synthetic data results: different selection methods results*

	Combination of all the previous methods		
Ranking method	Maximal number of occurrences	Maximal product of probabilities	Maximal average of probabilities
33.00	19.33	97.33	49.33
31.00	18.00	98.00	46.67
31.67	16.33	99.67	46.33
29.67	16.33	98.33	45.67
33.67	19.67	99.00	50.33
29.67	18.00	99.00	48.33
32.67	18.33	97.67	49.33
32.67	19.67	97.00	50.00
34.33	21.00	98.00	49.00
29.67	18.67	98.00	48.00
31.90	18.53	98.20	47.99

Table 3.4. *Synthetic data results with other selection methods*

5%: execution time	results	50%: execution time	results	90%: execution time	results
640ms	99.67	828ms	97.67	812ms	98.67
781ms	98.67	968ms	97.33	1s	99.67
641ms	98.00	750ms	99.00	844ms	98.33
703ms	97.33	813ms	98.67	969ms	97.00
828ms	98.67	859ms	98.33	922ms	98.33
734ms	98.67	828ms	99.33	891ms	97.67
750ms	98.33	875ms	98.67	922ms	98.33
750ms	99.00	781ms	99.00	890ms	99.33
688ms	98.67	797ms	97.00	859ms	99.00
750ms	97.67	844ms	98.67	890ms	98.67
726.5ms	98.47	834.3ms	98.36	899.9ms	98.5

Table 3.5. *Comparison between 5%, 50% and 90% attributes possibilities*

We can see in the tables 3.3 and 3.4 that the selection method is very important to have good results. The most accurate selection method is the maximal probability of the average probability with a 98.89 % accuracy and an execution time (for the forest construction) is about 733ms (with a maximum of 765ms), instead of 99 % of accuracy but an 840ms execution time for the Weka software.

The table 3.5 can also show that for a few calculus of set of attributes possibilities, we have nearly the same results, though we gain a lot of time when we don't calculate the whole possibilities.

Many tests have been made in such this way to see what the best combination was. As a conclusion, the thing is that increasing a lot the number of trees will not help having better results, but increasing the number of children or the maximal depth when there is a lot of data is important. Finally, taking 5% of attributes combination possibilities is quicker and give almost the same results.

Real descriptors

The results were not as good as excepted. The problem was that for the real descriptors, the features values are quite the same from one class to another one, and there are less information for each class, whereas there are much more classes. Consequently, the first results were about 15% which was not sufficient. The number of descriptors (and number of its features) have been increased, and the actual results are 42% for 2s 600ms execution time. The file is composed of 55 descriptors per class and 84 features to describe each one. Bigger data set has been tried to be launch, but it was too slow, so that the same data set has been tried with Weka, which, several seconds after, has stopped and said that it could not do it because it did not have enough memory (it was at 128 Mb of memory).

3.3.2 Encountered difficulties

At the beginning, an open source had been found, which created random trees from a given data set. However, the program was too static for us, because we could not change some parameters such as the number of children of a node, and so test many combinations in order to find the best one. Consequently, we thought about modifying it in order to have a more dynamical program. However, the lack of comments and documentation about the code rendered the modification impossible. Consequently, we decided to create ours.

Moreover, as the random forest is a big algorithm, and as there was not so much indication about how to construct it, it was difficult to understand how it was working, and many times my responsible and I had a different point of view on it.

Then, to improve the accuracy of the random forest, we decided to use the information gain calculus, in order to have the most informative trees. To verify it, we thought about calculate the forest strength and the correlation (also called complementarity) between the trees [9], but we did not see the relation between the correlation and the accuracy of the tree. Moreover, the information gain calculus has been removed, because the results were not logical. Normally, the more attributes possibilities we selected, the better should have been the results. But it was not the case. For example, taking 5% of attributes possibilities and taking the attributes combination with the greatest information gain would have better results than taking 100 % of attributes possibilities. And taking 5% of attributes possibilities and then the attributes combination with the most little information gain would have better results than taking 50% of them but less than 100% of them.

Finally, as I was not working on the generation of real descriptors, I had to wait that this part has been developed, and as my colleague had a lot of work, I had nearly no work(except writing the report and trying to improve the algorithm) during one month and a half. Then, when I got the real data, the results were not as excepted, so that I look at the code more carefully and find some errors in my code, which changes all. These problems may have been resolved with one month and a half more, which correspond to the time while I was waiting for the real data sets.

3.3.3 Eventual limits

A consequent problem is about the accuracy of the forest. As we say, for the moment, the results are only 40% accuracy, which is still not enough. Moreover, the more information we have, the longer is the execution time, which will be a problem in the future, as we are working in a real time environment.

3.3.4 What the company can do with it

The company think that the results can be improved, so they will continue to work on it and improve its accuracy and its execution time. An article would be written in the month of September, if they can improve it before.

3.3.5 Suites and suggestions

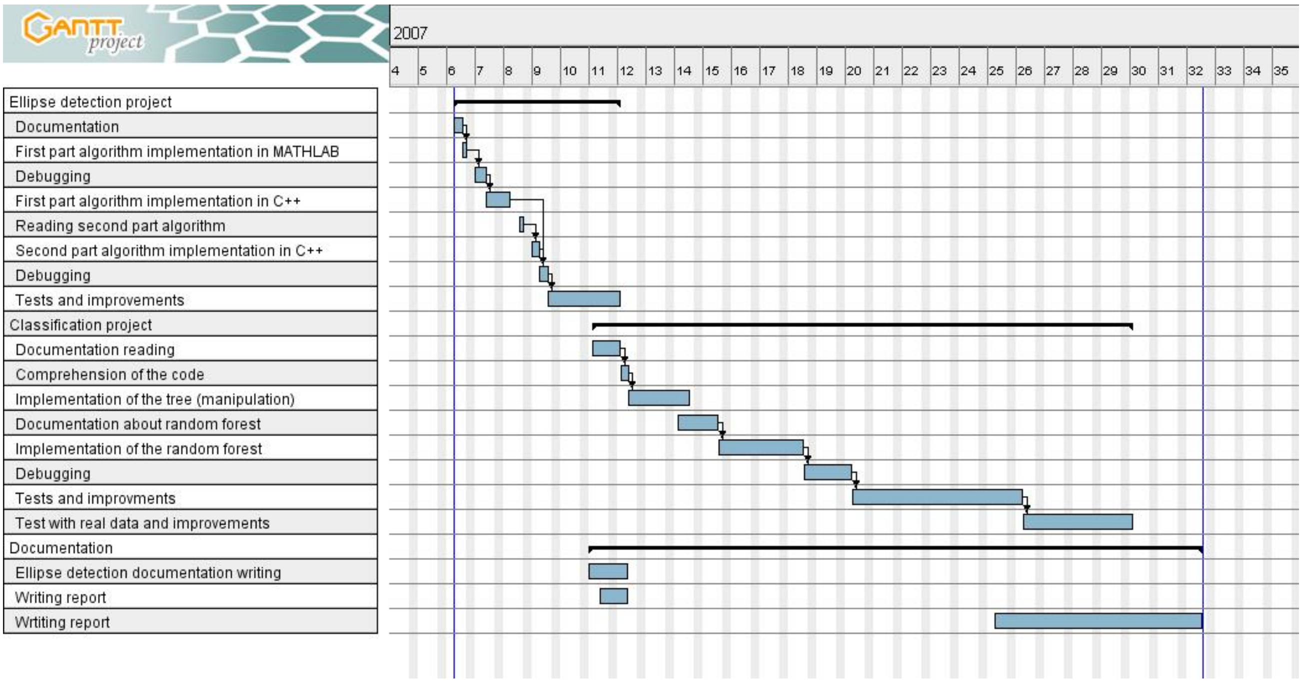
The information gain calculus was a good idea to detect the best attributes combination, but was not working. A research has to be made on this, and see where the problem was, or why it cannot work. However, if the attributes combination selection is improved, it would mean that to improve the results, we will have to calculate more attributes combinations possibilities. Consequently, an optimization has to be made. This could be done by looking at the code carefully and see what can be deleted.

Chapter 4

Conclusion

4.1 Real planning

The ellipse detection has been realized in one month and a half, and 3 months have been used to develop the classification program. The rest of the time was centered on the documentation writing. More explanations are given in the Figure 4.1.



Ellipse detection project
Documentation
First part algorithm implementation in MATHLAB
Debugging
First part algorithm implementation in C++
Reading second part algorithm
Second part algorithm implementation in C++
Debugging
Tests and improvements
Classification project
Documentation reading
Comprehension of the code
Implementation of the tree (manipulation)
Documentation about random forest
Implementation of the random forest
Debugging
Tests and improvments
Test with real data and improvements
Documentation
Ellipse detection documentation writing
Writing report
Wrtiting report

Figure 4.1. *Internship planning*

4.2 Contribution to the company

The company that I worked very well and was very efficient. They have appreciated my independence, in the sense that based on a few recommendations, I could start working on my own, looking for additional information, updates and references, most of them being successful and of interest. The classification module based on random trees I have been implementing works well. They think that they will be able to use my module in a practical application.

Apart from work, I have integrated myself perfectly and the fact that I already spoke Spanish has made it easier. In general the researchers of my group are happy with her. They would like her to stay with us, but for personal reasons, I probably will go back to France.

4.3 Technical encountered problems

The main technical problems that I had came from the fact that I did not know well some of the software (or libraries) that I used. So that it took me some days to learn how they worked. As in MATLAB, for example, the arrays were from 1 to n, instead of from 0 to n-1. I also had to be careful about the initialization of the variable in MATLAB, because when you declare it once, even if you stop the algorithm, it keeps the variable (and with its old values) in memory.

Another problem was because of the use of MFC. In fact, even if many things are very simple to create, as we can use an HCI to do it (we don't have to program it, as it is the case for WxWidget for example), many constraints rendered its use very difficult. For example, you had to put chronologically the buttons in order to pass the tabsteps correctly. Consequently, if you wanted to put a new button in the middle of your IHM, you had to delete all the previous or all the last buttons. This was very annoying and made loose a lot of time.

Then, in order not to have deleting problems with the pointers, we decided to use adaboost's smart pointers. As it was the first time I used it, I had difficulties to understand how it was working. For example, if you wanted to create a new pointer, you had to write this line :

4.4 Personal contribution

Technically, I have learnt a lot with them. At the beginning of my internship, I did not know anything of classification, and just a few about recognition patterns, and now I am controlling quite a lot of it.

Moreover, I have seen how was working a research centre, even if it this centre is a little bit different from the French research centers, as they make a technological bridge between the universities and the companies. This experience is so rewarding that I really want to continue working in this domain. That is why I applied for imagery masters and also PhD, but CIFRE ones, because it could allow me to be connected to companies still, as in here.

As I was for the first time in the basque country, I visited a lot this region. I also talked to the basque people about their culture, their history, their way to think, how they consider French people, and I also learnt Basque (but only expressions and some words). The people here are very open-minded, and it was a great pleasure to talk to them.

Finally, even if there were many Spanish people, there were also Colombian and German people. And as some German people did not speak Spanish, I always had to speak in English with them. At first, it was pretty difficult, because I had to think in French, and then traduce it in English or in Spanish, so that makes it very confusing some times, but now I have no problems to speak either in Spanish or in English or French.

4.5 School contribution

The school contribution has been seen in many points. At first, I would say that all I have learnt about C++ was really useful. In fact, as I had to develop in this language and as I had a good knowledge of it, it took me much less time than if I had to learn it by myself. Then, about the way to work, because I had to make a lot of research, and also to know how to begin and correct my faults.

As I had to speak English many times, the fact that have kept studying English during these years were very useful.

Finally, as for this year I worked on the ellipses detection during my study, in a practical work, I knew well the problems and difficulties there was.

4.6 Conclusion

As a conclusion, I would say that this experience was extraordinary, and I really enjoyed it. I would never regret this experience, and all of what it has brought me. I think finishing my studies on such a great internship is very good. In fact, when I arrived at VICOMTech, I was pretty sure that even if they proposed me a job there, I would not stay, because I wanted to work in France after and not in Spain. But after having worked with them, seen their kindness, on what they were working and also all the advantages of staying there, I really hesitated, and that was really hard for me to say no. I hope that I will keep contact with them, in the future, because they have many interesting projects and ideas, and I think it could be a real asset to work with them.

Bibliography

[1] Wikipedia

[2] H. Talbot and B. C. Appleton: *Elliptical distance transforms and the object splitting problem*; (2002)

[3] Ludmila I. Kuncheva: *Combining Pattern Classifiers, Methods and Algorithms*; sections 1.1-1.6, 2.4; pages 1-36, 68-70; (2004)

[4] J. M. Prez de la Fuente: *Árboles Consolidados: Construcción de un árbol de clasificación basado en múltiples submuestras sin renunciar a la explicación*; chapters 0-2; (January 2006)

[5] L. Breiman: *Random forest*; Machine Learning, 45; 532; (2001)

[6] V. Lepetit, P. Lagger and P. Fua: *Randomized Trees for Real-Time Keypoint Recognition*; (June 2005)

[7] V. Lepetit and P. Fua: *Keypoint Recognition using Randomized Trees*; (January 2006)

[8] B. Wilson: *Induction of Decision Trees*; (2006)

[9] L. Bovino, G. Dimauro, S. Impedovo, G. Pirlo and A. Salzo: *Increasing the Number of Classifiers in Multi-classifier Systems: A Complementarity-Based Analysis*; sections 2, 3; (2002)

[10] F. Boto Sánchez: *Clasificación Múltiple para Dígitos a Máquina de baja calidad. Desarollo de un Prototipo de Experimentación*; pages 30-36; sections 2.5.2, 2.5.3; (June 2006)

[11] Fuad M. Alkkot and J. Kittler: *Improving the Performance of the Product Fusion Strategy*; 0 section 2; (2000)

[12] L. A. Alexandre, A. C. Campilho, M. Kamel: *On combining classifiers using sum and product rules*; (2001)