

École polytechnique de Louvain

Segmenting letters in words written on a tablet during a dictation

Author: **María EUGENIO CONCHA**
Supervisor: **Christophe DE VLEESCHOUWER**
Readers: **Simon CARBONELLE, Marie VAN REYBROECK**
Academic year 2018–2019
Master [120] in Electrical Engineering

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
ÉCOLE POLYTECHNIQUE DE LOUVAIN (EPL)
MASTER [120] IN ELECTRICAL ENGINEERING
Master Thesis

SEGMENTING LETTERS IN WORDS WRITTEN ON A TABLET DURING A DICTATION

MARÍA EUGENIO CONCHA

2019

Acknowledgments

Life consists of a series of stages and with this “Trabajo Fin de Máster”, “TFM”, “Master Thesis” or “Mémoire” I finish my stage as a student to make way for a new one as Ingeniera de Telecomunicación or Telecommunication Engineer. I would not dispute that these two years doing the Master both in Madrid and in Louvain-la-Neuve did not be, as it is said in Spanish, “un camino de rosas”, but apart from studying a lot, I have met very special people for me who have made these years more enjoyable. For that, I want to thank them, along with the people who have continued to be by my side.

To my parents, because they are always showing me their support (and giving me envy by sending photos of homemade food) even if they are 1593 kilometers away.

To my friends of Fuenlabrada, for enduring my sorrows with my career and with my life in general (and for being the best).

To my friends at the UPM, for enduring my perfectionism in the reports (I know that is not easy) and for all the laughter we have thrown together.

To “les meilleurs cokoteurs du monde”, because of the motivational coffees to cope better with the day and for being the future writers of my biography.

To my friends of LLN, because of all the “calvarios” that we have lived (which are not few), because of all the dinners, because of those special moments of intense “TFMeo” in Candix (while I am writing this some of you are here with me) and because “nenes, os pasáis”.

Finally to thank Christophe de Vleeschouwer for all the help received during the development of this thesis, Clara Sansalvadó, because without her work this thesis would not have been possible and the members of the jury, Simon Carbonelle and Marie Van Reybroeck for evaluating this work and for being part of this research.

Gracias a todos.

Thanks to everyone.

Merci à tous.

María Eugenio Concha

Abstract

The goal of this Master Thesis is, given writing data recorded by a tablet during a dictation task as a temporal sequence of samples that measure, as a function of time, the spatial position of the pen-tip on the tablet, as well as “Pen-Up” and “Pen-Down” switching, be able to segment that data into smaller units, starting from a higher level, like lines of text, until a lower level, the letters.

On the one hand, given a series of samples corresponding to texts written by children, a temporary analysis of these samples is carried out in order to segment the text into lines. With this lines of text, the links between strokes are analyzed and classified into two groups, inter-letter and intra-letter links, by means of a binary classifier based on SVM with the aim of segmenting each line of text into words.

On the other hand, to perform the segmentation of words into letters, in addition to using the data provided by the tablet, it is used the position of all the templates given by a method implemented in a previous Master Thesis [1] that first warps reference letter templates using Lucas-Kanade optimization algorithm and then defines an optimal sequence of reference templates and warping parameters, based on a shortest-path algorithm. With this, it is defined a minimization problem which finds the best set of endpoints of the letters which minimizes the distance between the warped templates and the function of time that defines the tracing of the word in order to segment the word into letters. It is defined two different strategies to perform that, one based on the use of the center of each template and another on the used of the distance map of each template.

Finally, an analysis of the results of both word and letter segmentation will be presented as well as an analysis of the weak points that the proposed methods have.

Keywords Children handwriting, cursive writing, HWR, letter segmentation, Lucas-Kanade algorithm, shortest-path algorithm, SVM, template warping, word segmentation.

Contents

Acknowledgments	i
Abstract	iii
List of abbreviations	xv
1 Introduction	1
1.1 Motivation	1
1.1.1 The importance of handwriting	1
1.1.2 Handwriting during the childhood	2
1.1.3 Handwriting and learning difficulties	2
1.2 Research statement	4
1.3 Goals	4
1.4 Methodology	4
1.5 Organization of the thesis	5
2 State of the art	7
2.1 Handwriting recognition	7
2.1.1 Online handwriting recognition	8
2.1.2 Offline handwriting recognition	11
3 Word and line segmentation	13
3.1 Page segmentation	13
3.2 Line segmentation	15
3.3 Word segmentation	16
3.3.1 Feature Extraction	17
3.3.2 Classification of features using SVM	18
4 Warping letters in a word overview	23
4.1 Lucas-Kanade algorithm	24
4.1.1 Image and distance map generation	24

4.1.2	Analysis area definition	25
4.1.3	Template warping using Lucas-Kanade algorithm	26
4.2	Graph based letter alignment	27
4.2.1	Building the directed graph	27
4.3	Results of the method	30
5	Letter segmentation	31
5.1	Data preparation	31
5.2	Distance between data and templates	34
5.2.1	Using the center of the template as reference	35
5.2.2	Using the distance map of the template as reference	36
5.3	Calculation of the endpoints of the letters	39
5.3.1	Solving the cost function	41
5.4	Special cases	44
5.4.1	Special characters	44
6	Results	49
6.1	Word and line segmentation	49
6.1.1	Definition of Ground-Truth	49
6.1.2	Data used	49
6.1.3	Metrics used	50
6.1.4	Word segmentation results	50
6.2	Letter segmentation	52
6.2.1	Definition of Ground-Truth	52
6.2.2	Metrics used	53
6.2.3	Letter segmentation results	55
6.3	Weak points	57
6.3.1	Detection of omitted letters in a word	57
6.3.2	Detection of overlapped letters	62
6.3.3	Similarities in the appearance of two letters	63
7	Conclusions and further word	65
7.1	Conclusions	65
7.2	Further work	66
A	Data used	67
A.1	Texts data	67
A.2	Words data	68
A.3	Reference data	68

B	Impacts	71
B.1	Introduction	71
B.2	Relevant impacts	71
C	Economic budget	75
C.1	Direct cost	75
C.2	Other costs	76
C.3	Total cost	77
	Bibliography	78

List of Figures

2.1	Letter “A” represented in: left) offline HWR, right) online HWR. Image from [2]	8
2.2	Steps of HWR. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.	8
2.3	Top) types of slope, bottom) types of slant. Image from [3].	10
2.4	Different styles of writing. Image from [4].	10
3.1	Diagram of the process of obtaining words from a text recorded by the tablet. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.	13
3.2	Example of superposed text recorded by the Wacom [®] tablet.	14
3.3	Evolution of the a) X data, b) Y data, along the time.	15
3.4	Pages obtained after the page segmentation procedure.	15
3.5	Evolution of X data of the a) first page, b) second page, along the time and the position of the beginning of the lines.	16
3.6	Two first lines of the first page.	16
3.7	Strokes which form the phrase “ <i>du zoo</i> ”.	16
3.8	Diagram of the process of obtaining the prediction of the off-strokes based on a SVM. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.	17
3.9	Terms used to define the strokes in the line “ <i>du zoo</i> ”.	18
3.10	Representation of the dataset.	19
3.11	Division of data when it is used 4-fold cross validation.	20
3.12	Average of the performance metrics of the SVM after executing it 100 times for different number of features.	21
4.1	General scheme of the method for warping letters in a word. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.	23

4.2	Outline of the process of obtaining the templates. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.	24
4.3	Estimated positions of the letters of the word “glace”.	25
4.4	Distance map of the written word “glace”.	26
4.5	Estimated position and AA of letter “g” in the word “glace”.	26
4.6	Best template matching result using LK algorithm and its corresponding warped template.	27
4.7	Outline of the graph based letter alignment method. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.	27
4.8	Representation of the search window, SW.	28
4.9	Left and right points of the template of the letter “o”.	28
4.10	Cost function of: a) NCC and b) inter-letter distance. Image from [1]. . . .	29
4.11	Graph representation for finding the best positions of the letters in the word. Image from [1].	30
4.12	Comparison between: a) the estimated position of the letters and b) the position of the letters obtained after applying LK algorithm and the graph-based letter alignment.	30
5.1	General scheme of letter segmentation. Notation: the oval blocks indicate input or output data, of these, the greens indicate that the data is in the coordinate system of the tablet and the oranges in the image domain, and the rectangular blocks indicate processes.	32
5.2	Representation of the word “glace” in left) image system and right) tablet domain.	33
5.3	Systems of coordinates used. Left: Coordinates of a template system (U,V). Right: tablet coordinates system, (X,Y).	33
5.4	Parameters which define the image in the template domain (left) and in the tablet coordinates (right).	34
5.5	Reference parameters to take into account to make the value change of the Y axis.	35
5.6	Tracing of the word “glace”, the position of the templates and their corresponding centers.	36
5.7	Distance graph between the center of templates of the word “glace” and each point of the tracing.	36
5.8	Top: Letter templates of word “glace”. Bottom: Distance map of the templates of word “glace” after applying Lucas Kanade.	37
5.9	Distance map of the letter “c” above the tracing of the word.	37
5.10	Global and local coordinates of a template.	38

5.11	Different cases of positions that the points of the curve $f(t)$ can have with respect to the DM (represented as a blue rectangle).	38
5.12	Distance graph between the distance map of templates of the word “glace” and each point of the drawn.	39
5.13	Shortest-path algorithm diagram.	42
5.14	Position of the final endpoints of the word “glace” in the distance graph when: a) using CT as reference, b) using the DM as reference.	43
5.15	Position of the final endpoints of the word “glace” in the tracing of the word when: a) using CT as reference, b) using the DM as reference.	43
5.16	Distance graph of “image” without taking into account the point of the “i”.	44
5.17	Result of segmenting the word “image” where the point of the “i” appears with the “e”.	45
5.18	Outline of the different cases of endpoint examination that a word can have.	45
5.19	Distance graph of “image” taking into account the point of the “i”.	46
5.20	Distance graph and endpoint positions after applying the algorithm to the word “très”.	47
6.1	ROC of the a) first configuration, b) second configuration.	52
6.2	Results after performing the word segmentation. Top) GT . Bottom) Prediction.	52
6.3	Possible set of endpoints of each letter in the word “glace”.	53
6.4	Possible set of endpoints of each letter in the word “image”.	53
6.5	Representation of $\Delta t_{I_{start}}$ and $\Delta t_{I_{end}}$ in the first letter of “glace”.	54
6.6	Samples error of using CT and of using DM as reference: a) absolute error, b) relative error.	55
6.7	Distance error of using CT and of using DM as reference.	56
6.8	Tracing of word “bravo”: a) all the samples recorded by the tablet, b) strokes of the word.	56
6.9	AA area and best template position after applying the LK algorithm in: a) “glace” when the original word is “glace”, b) “glace” when the original word is “glrace”.	59
6.10	Curve map of the NCC when the template is placed in the allowed positions inside the AA when the original and the written word are “glace”.	59
6.11	Curve map of the NCC when the template is placed in the allowed positions inside the AA when the original word is “glrace” and the written one is “glace”.	60
6.12	Probability density functions in the case of a) the original and the written word is the same (“glace”), b) the original word is “glrace” and the written one is “glace”.	60

6.13	New cost function for the NCC.	61
6.14	Example of overlapped letter where a part of the letter “b” of “chambre” is rewritten: a) image of the word “chambre”, b) its associated distance graph using the <i>CTs</i> as reference.	63
6.15	Final position of the letters obtained after applying the LK algorithm and the graph-based letter alignment to the word “mulo”.	63
A.1	Reference letters. Image from [5].	69

List of Tables

6.1	Confusion matrix for: left) 4 features, right) 9 features.	51
6.2	Results of the SVM model.	51
6.3	Costs given by the shortest-path algorithm when the original word and the written one are the same.	58
6.4	Costs given by the shortest-path algorithm when the original word and the written one are not the same.	58
6.5	Costs given by the shortest-path algorithm when the original word and the written one are the same using the new cost function.	61
6.6	Costs given by the shortest-path algorithm when the original word and the written one are not the same using the new cost function.	62
A.1	Description of the sentences' data recorded by the.	68
A.2	Description of the words data recorded by the tablet.	69
C.1	Description of the material cost of the project.	75
C.2	Labor cost of the project.	76
C.3	Summary of the economic budget of the project.	77

List of abbreviations

Abbreviations

- AA: Analysis Area.
- AUC: Area Under Curve.
- CT: Center of Template.
- DM: Distance Map.
- FN: False Negative.
- FP: False Positive.
- GT: Ground Truth.
- HWR: Handwriting Recognition.
- I: Image.
- LK: Lucas-Kanade.
- NCC: Normal Cross Correlation.
- NSP: Non-Separation Point.
- OCR: Optical Character Recognition.
- ROI: Region of Interest.
- SP: Separation Point.
- SSD: Sum of Squares Differences.
- SVM: Support Vector Machine.
- SW: Search Window.
- T: Template.

- TN: True Negative.
- TP: True Positive.

Notation

- S : number of points (samples) that the function which defines the tracing of the word has.
- N : number of letters of the word.
- E : number of nodes in a graph.
- C : number of “special characters” in a word.
- C' : number of “special characters” that are not placed in the last position of a word.
- $g(t)$: function which defines the tracing of the word.
- $f(t)$: transformed function of $f(t)$.
- $\xi(t)$: distance function.
- (X,Y) : definition of global coordinates in the tablet domain.
- (X',Y') : definition of local coordinates in the tablet domain.
- (U,V) : definition of global coordinates in the image domain.
- (U',V') : definition of local coordinates in the image domain.
- (x,y) : coordinates of a point in the tablet domain.
- (u,v) : coordinates of a point in the image domain.
- m_u : magnitude added to the width of the template to establish the AA.
- m_v : magnitude added to the height of the template to establish the AA.
- r_u : translation of a point horizontally.
- r_v : translation of a point vertically.
- s_u : scale factor of the width.
- s_v : scale factor of the height.
- Δx : horizontal distance between the Y axis and a point.
- Δy : vertical distance between the X axis and a point.

-
- Δu : horizontal distance between the V axis and a point.
 - Δv : vertical distance between the U axis and a point.
 - δx : horizontal distance between two points.
 - δy : vertical distance between two points.
 - W : width of the ROI in the tablet domain.
 - w : width of the ROI in the image domain.
 - H : height of the ROI in the tablet domain.
 - h : height of the ROI in the image domain.
 - w_t : width of the template.
 - h_t : height of the template.
 - e : endpoint, defined as a timestamp.
 - $\mathbf{e}=(e_0, e_1, \dots, e_N)$: set of endpoints of the letters which form the word.
 - $\mathbf{e}^u$: vector which contains the unvisited nodes.
 - $\mathbf{e}^v$: vector which contains the visited nodes.
 - $\hat{e}$: current node.
 - $\mathbf{e}_{\text{intersect}}$: vector which contains all the intersection points between two given distance curves.
 - $\mathbf{C}$: vector which contains the curves associated with the “special characters”.
 - $\mathbf{D}$: vector which contains the distances between the initial node and the rest of the nodes.
 - $D_{(a,b)-(m,n)}$: distance between the a -th endpoint of the b -th set and the m -th endpoint of the n -th set.
 - $d(a,b)$: distance between a and b .
 - $\mathbf{p}$: vector which contains the set of warping parameters.
 - $I(\mathbf{W}(\mathbf{u}; \mathbf{p}))$: warping applied to the image I under the parameters $\mathbf{p}$.
 - Δt : number of samples between an endpoint and the GT.

Chapter 1

Introduction

1.1 Motivation

1.1.1 The importance of handwriting

The writing is an important task for human communication. It serves as an instrument of social interrelation by serving as the means by which a person communicates with others. Writing also serves as a form of expression, to express in this way everything that is felt in the inner self and can not be expressed orally.

During the childhood, writing is not only a cultural and social tool but also a complex cognitive-symbolic activity that involves the construction of meanings by children and impacts on their intellectual development by involving higher cognitive and metacognitive processes [6].

Writing is a linguistic motor task which has two different stages, spelling and motor production. The first one is referred to the central processing and the second one to the peripheral processing.

The spelling is referred to as central processes because it involves the selection and activation of the different orthographic representations. For many researches, the spelling process is completed before the motion production, appearing a temporal lapse between the two stages called latency. On the other hand, the motor production is referred to peripheral processing and its investigations are related to the selection and activation of motor programs which contain information about what is being written, such as the letter shape [7].

Currently, there are some researches that are studying the interaction between central and peripheral processing. These cover some aspects of writing like the number of occurrences of a word during the handwriting and the orthographic regularity in the spelling of the words. To study that, some researchers [7] do different experiments using both regular and irregular words during a dictation task, understanding regular words as those that

refer to the possibility of spelling a word correctly applying the most phoneme-grapheme conversion rules. The results show that the latencies are longer in words that are both irregular and infrequent, which can indicate that the spelling processes were active before starting to write. Writing duration indicates the engine processing during movement production. This means that not only the way in which words are written but also the duration, are determinant to analyze different cognitive aspects of people.

1.1.2 Handwriting during the childhood

The characteristics which define quality handwriting are legibility and speed which are not very common in children's handwriting. Graphomotor skills, visual-motor integration and fine motor writing are terms which are used for evaluating writing readiness. For young children, the development of handwriting is a complex task that requires the coordination between cognitive, motoric and neuromotor processes while for most adults this task may feel like an automatic process [8]. When children begin to experiment with writing at a young age, they draw meaningless scribbles, although some studies [9] show that those scribbles contain typical characteristics of conventional writing such as linearity and directionality. Then, they start to write copying geometric shapes like vertical and horizontal strokes and circle.

Another point to take into account during the children's writing is the development of the motor skills which allow them to manipulate different objects with their hands, which impedes their ability to produce letters accurately. To write the letters correctly children need to have a complete visual representation of each letter, recognize the line segments which compose a letter and to have the ability to reproduce the sequence and the direction with the segments which form the letter [8]. Then, they can develop more complex writing tasks such as dictation when the recognition of the spelling is a prerequisite [10].

1.1.3 Handwriting and learning difficulties

Dyslexia is a neurological based disability characterized by a specific learning difficulty with written language. Some researches demonstrate that people with dyslexia have difficulties with reading acquisition and accurate spelling [11]. But this is not the only learning difficulty that can affect written language. Other problems such as dysgraphia affect notably writing. Some of the most relevant conditions of writing that can be affected by the different learning difficulties are summarized below [12].

- **Dysgraphia:** it affects the ability to write efficiently and effectively. The main problems are in the control of the pencil, in drawing letters and in placing spaces between letters and words which cause that the writing is disorderly and slow. Children with dysgraphia may not be able to write as fast as they think which makes it difficult to put ideas of the paper in an organized way.

- **Dyslexia:** it generates difficulty with reading and spelling and it can also affect writing and speech. Problems in identifying individual sounds in words make it difficult to decide which letters are necessary to write the word. The main characteristics that people with dyslexia present in writing are the bad spelling, punctuation and capitalization and the incorrect sequencing of ideas.
- **Difficulties of expressive language:** it creates problems to express thoughts and ideas using spoken language, children with this condition may have problems in both the training and the production of language. They also have difficulty expressing themselves through writing using a limited vocabulary, using indefinite words, presenting difficulties to produce complex sentences, omitting some words and confusing grammatical tenses.
- **ADHD (attention deficit hyperactivity disorder):** it may involve problems to focus, act without thinking and be extremely active, the distraction can cause the writing consumes a lot of time and that ideas can be omitted. Moreover, writing can show a limited vocabulary and a simple sentence structure.
- **Difficulties of visual processing:** it causes difficulty in understanding the information that the brain receives through the eyes, however, these difficulties are not a problem of sight. The visual processing is related to remembering the shape of the letters, estimating the spaces on a page and planning how many words fit on a line. This has some consequences during writing such as poor letter tracing, reversing the letters (write the letters upside down), keeping an inappropriate space between words and letters, skipping lines on paper, writing beyond the margin lines and it can produce inaccuracies when copying.
- **Dyspraxia:** it makes it difficult to plan, coordinate and execute motor movements. This can affect the ability to use a pen or a keyboard correctly which can cause writing to be frustrating and messy. The main characteristics of writing of people with dyspraxia are that letters are made poorly and with variable size, there are inadequate spaces between letters and between words and the writing can be too marked or too tenuous.

As it can be shown previously, people that suffer some type of learning difficulty present certain characteristics in their writing. These characteristics can be grouped into three different groups:

- On the one hand, are those related to the shape of the letter and the spacing between words and between letters.
- On the other hand, there is the speed of writing, related to the time in which the affected person takes to process their ideas and translate them on the paper.

- Finally, there are some characteristics related to the content of the written text: if the words are repeated, if complex structures are used in the sentence, if the correct tenses are used, etc.

1.2 Research statement

This project is a part of a research carried out by the Psychological Sciences Research Institute and the Institute of Information and Communication Technologies, Electronics and Applied Mathematics, both from the Université Catholique de Louvain. The goal of the research is to gain a better understanding of what constitutes the graphic complexity of written material and to understand cases of children with dyslexia [5].

The working data, which is explained in Appendix A, was written by children in a dictation task using a digital tablet with a digital pen. They simply had to write down the words they heard.

1.3 Goals

Based on the data given for analyze the writing of children, this thesis has two main goals. On the one hand, as some of the data consists in several phrases writing by different children, one objective is to segment a given handwritten text firstly into lines and then into words doing a classifier based on support vector machines (SVM). On the other hand, understanding the method proposed by Clara Sansalvadó [1] which consists in obtaining the best position of the letters in a word using Lucas-Kanade algorithm together with a graph based letter alignment, and based on her method, design a procedure of segmentation of words into letters.

1.4 Methodology

To achieve the goals of the project, it was used the following methodology.

- Word segmentation: using methods based on the understanding of the data provided by analyzing it and to design a classifier based on support vector machines (SVM) that allow to segment a line of text into different words.
- Letter segmentation: based on the methodology proposed by C. Sansalvadó [1], performing the segmentation of words into letters using a shortest-path algorithm by defining a minimization problem.

1.5 Organization of the thesis

This thesis is structured, including this introductory chapter, in six chapters and three appendixes:

In the **Chapter 2** it is presented the state of the art corresponding to an introduction of handwritten recognition (HWR) explaining the two different types which exist.

In the **Chapter 3** it is described the method proposed to perform the line and the word segmentation from a given text recorded by the tablet.

In the **Chapter 4** it is explained the Lucas-Kanade template warping with the graph-based letter alignment used for obtained the best position of the letters in an image of a word.

In the **Chapter 5** it is described the proposed strategy to perform the segmentation of the words into letters based on the Lucas-Kanade template warping.

In the **Chapter 6** it is presented the results obtained of applying both letter and word segmentation as well as the weak points found during the development of the project.

In the **Chapter 7** it is summarized the conclusions obtained and it is presented an analysis of the improvements that can be done as a further work based on this project.

Finally, in the **Appendix A** it is described the database employed to perform the project, in the **Appendix B** the impacts that this project can have is explained and in the **Appendix C** it is summarized the economic budget of developing this project.

Chapter 2

State of the art

In this section the state of the art of handwriting recognition is presented. The different types of handwriting recognition that exist today are explained as well as the applications that they have at present.

2.1 Handwriting recognition

Handwriting recognition or HWR is the ability of a computer to receive and transform handwritten inputs, such as photos or information collected by a touch-screen, into digital information. The handwritten data possess different characteristics which make it a rather complex task to carry out such as the writing style which varies notably between (i) individuals because of the preference of writing or because of the age and location and (ii) the same person because of the speed, writing surface, the mood of the person during the writing, etc [13].

There are two types of handwriting recognition depending on the nature of the input data that will be analyzed. On the one hand, there is the online handwriting recognition which consists in different methods that deal with the automatic processing of a message that is written using a digital device that captures information about the drawing of the pen-tip. This information can be the “Pen-Up”/“Pen-Down” switching, the position or the pressure as a function of time, for example. On the other hand, there is the off-line handwriting recognition in which, contrary to what happens in the previous one, a document with handwritten text usually captured optically by a scanner is used as input. So, in this kind of handwriting recognition, the written text has been created before the processing [14][15]. HWR has a wide variety of applications such as bank check processing, automatic address reading or for the extraction of the content from digital databases [16]. Depending on the application, the recognition system is either based on scanned image data (offline) or on pen-trajectory data which is recorded during the writing process (online) as it can be seen in Figure 2.1 [2].

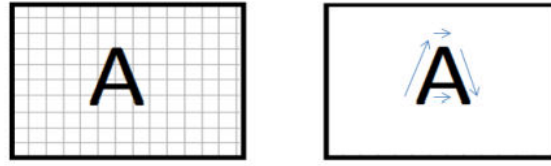


Figure 2.1: Letter “A” represented in: left) offline HWR, right) online HWR. Image from [2]

According to the nature of the data processed, each one of both types presents some advantages and disadvantages. One advantage of the online HWR is that spatially overlapping characters do not pose a segmentation problem since they are still separable by the time difference but by contrast, the temporal information can complicate the recognition since one letter can be written in different number of strokes (i.e., letter “E” can be written from one to four strokes).

In both cases, according to [15] and [2], a typical handwriting recognition system has the steps showed in Figure 2.2.

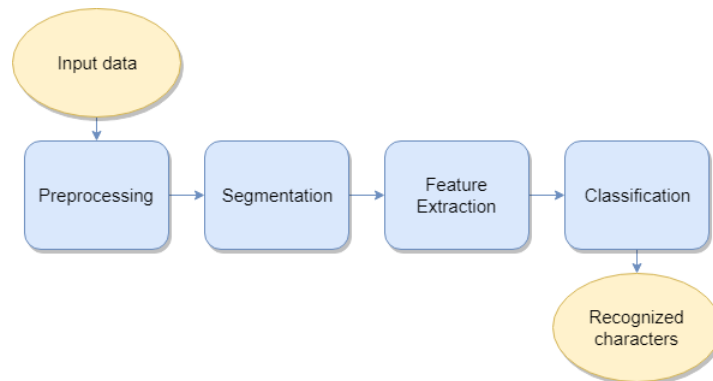


Figure 2.2: Steps of HWR. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.

2.1.1 Online handwriting recognition

The online handwriting recognition is the task of recognizing a text written in an electronic device with a pressure sensitive surface from which real-time information, such as the order of the stroke written by the user is obtained and preserved [2]. The data collection process is done automatically, and this data includes information related to the movements and the drawing performed during the writing with the pen-tip. This information involves different characteristics of the style of writing such as the pressure and, also, collects the time series (x,y) coordinates of the pen movement, pen-up/pen-down status and timestamp.

According to the applications that online HWR may have, in [14] three different groups

are distinguished: pen-based computers, signature verifies and developmental tools. The first one refers to the recognition of handwritten messages and gesture commands to interact with pen computing platforms. The second deals with a special type of text, the signatures, in order to verify the identity of the person. By last, the developmental tools exploit the neuromotor characteristics of handwriting to design systems for education and rehabilitation purposes.

To develop this type of HWR is necessary to have different elements that allow to collect and encode the data samples in real time during the writing task. For that, usually three elements are necessary:

- Pen.
- Electronic device with screen-touch.
- Specific software.

Methodology of online HWR

According to the outline presented in Figure 2.2 the methodology used in online HWR is the following [2][4][15].

- **Data acquisition:** This type of HWR requires a transducer which captures different characteristics of the tracing during the writing. To do that, the most common is the electronic tablet with its digital pen. This pen does two actions “Pen-Down” and “Pen-Up” and the trace between both is called stroke.
- **Preprocessing:** This phase is applied in handwriting recognition to remove the noise or some distortions as the irregular size of the text which can be presented in the input. Some of the steps are (i) noise removal, (ii) binarization, (iii) slant and slope correction and (iv) resize. In slant correction, it is computed some mean dominant vertically oriented slope, offsetting the slope of the data using the different between the computed slope and a slope of the vertical axis. The slope correction consists in computing the slope of the baseline and iteratively changing the position of the data in order to obtain a horizontal baseline. The middle texts of Figure 2.3 are examples of slant and slope correction.
- **Segmentation:** Is the process of dividing a given text into different units depending on the type of segmentation that is being carried out, which can be (i) line segmentation, (ii) word segmentation and (iii) character or letter segmentation. The segmentation task can be easier or more difficult depending on the type of the letter and the writing style. Figure 2.4 shows different styles of writing defined in [4].



Figure 2.3: Top) types of slope, bottom) types of slant. Image from [3].

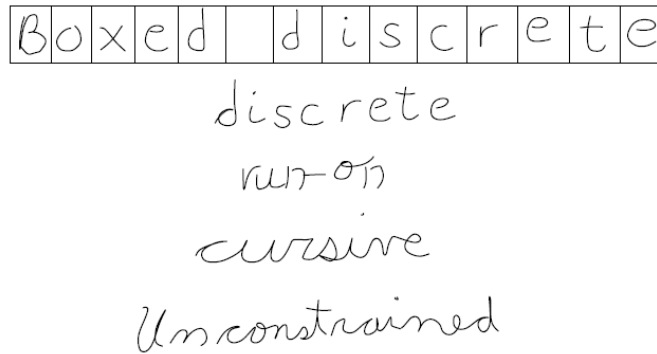


Figure 2.4: Different styles of writing. Image from [4].

In boxed discrete writing, the user writes inside a box, helping the task of segmenting and therefore, the recognition task. The discrete writing is also an easy type of writing to perform the HWR since the letters are written separately, to perform the segmentation it can be done a search of gaps between character along the horizontal direction and establish a threshold to classify the gaps into an inter-letter gap or into an intra-letter gap. The run-on and cursive writing are the more common type of writing and one way to perform the segmentation is by analyzing the strokes as the smallest unit of the word.

As said before, there are three types of segmentation:

- **Line segmentation:** this segmentation consists in divide the given text into different lines. In [17] it is proposed a method based on linear regression.
 - **Word segmentation:** consists in separate the text into different words. To perform that in [17] and [18] it is suggested a method based on the using of SVM.
 - **Letter segmentation:** in this case, the segmentation is performed with the goal of obtaining the letters from a given word.
- **Feature extraction:** The feature extraction is the process of collecting relevant information about an object that can describe it and which allows performing a

future classification based on these characteristics. Depending on the information of the objects needed, it can be divided the type of features into different groups:

- Statistical features: deriving from statistical distribution points. These characteristics take care of style variation and some of the main features are zoning characteristic loci, crossing and distance.
- Structural features: they are based on topological and geometrical properties of the character such as loops, aspect ratio, stroke direction, cross points, etc.
- **Classification:** As HWR is the task of transforming a written text into its symbolic representation like 8-bit ASCII character set, it is necessary to classify each segmented character in each corresponding representation. There are several classification methods.
 - Template matching models: in the training stage, these systems create templates of their elementary handwriting segments and represent them by the feature parameters used for recognition. After that, these systems measure the distance from the segment of data to the template segments and, for each segment, they generate a likelihood value which indicates how close the data is from each template.
 - Statistical models: One of the main statistical models is the Hidden Markov Models (HMM) which have been used in speech recognition. In HWR, different characters models are built at the training stage. With the training data a graph containing states and the probabilities associated with different state transitions.
 - Neural Networks models.

Finally, it is printed the corresponding recognized characters in the structured text form.

2.1.2 Offline handwriting recognition

The offline handwriting recognition uses as an input a scanned document or an image in which appears handwritten text. The main goals in this type of HWR are character (OCR) and word recognition [19].

Optical character recognition (OCR) is a procedure whose aim is to digitize texts which are identified from an input image that presents different symbols or characters belonging to a specific alphabet. The input image can be a scanned document, a photo of a document, a scene photo where some text appears, or text superimposed on an image. This technology allows recognizing automatically the different characters through an optical mechanism which works as the eyes. The brain understands the image seen by the eyes and it is able to recognize what it is seeing, identifying the characters. OCR aims

to work like the human brain, trying to understand a text, both handwritten and printed text [4].

The importance of the OCR is due to the multiple application that it has, such as automatic number plate recognition, passport recognition, make electronic images of printed documents searchable or converting handwriting in real time to control a computer [20].

Chapter 3

Word and line segmentation

In this chapter, it is going to be explained how to process a given text recorded by the tablet in order to segment it firstly in different lines and, then, into words. With the result of this segmentation, it can be applied the algorithm of letter segmentation explained in Chapter 4 and Chapter 5.

The outline of this chapter is presented in Figure 3.1 where the different stages from a given raw text to the obtaining separate words are shown.

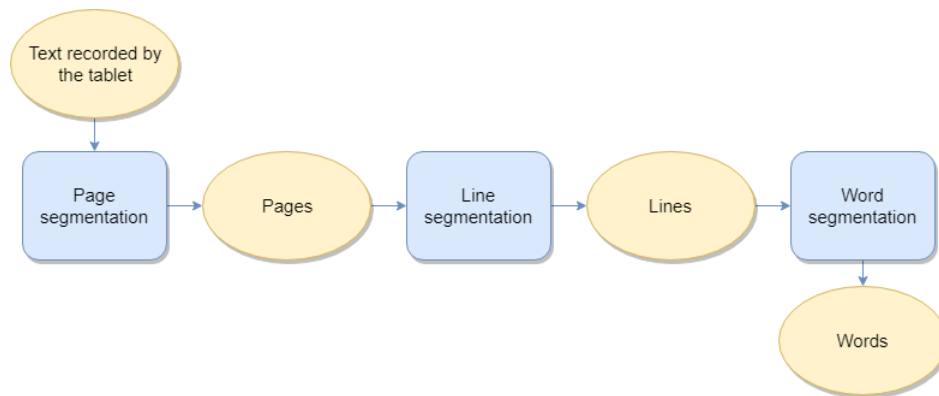


Figure 3.1: Diagram of the process of obtaining words from a text recorded by the tablet. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.

3.1 Page segmentation

The first step to perform the word segmentation is to divide the text recorded by the tablet into different lines. To develop that, it is considered the way in which the tablet recorded the data, that is, as explained in Appendix A, it is considered that the users write in a normal way of French writing, that means, starting from the upper-left corner and continuing to the right. When the page ends, Wacom[®] generates a new one and the text

written in this new page occupies the same position than the previous one. This means that when viewing the data collected by the tablet, there is superimposed data as it can be seen in Figure 3.2.

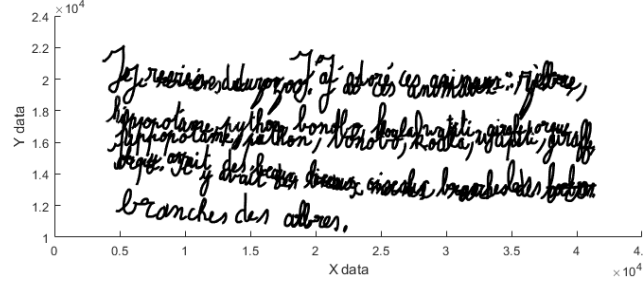


Figure 3.2: Example of superposed text recorded by the Wacom[®] tablet.

To solve this problem, it is performed a study of the different components of the data recorded by the Wacom[®] tablet, the X data and the Y data which in normal French writing will have the following appearance:

- **X data:** the data increases along the time until the line ends, forming a saw-tooth function where the local maxima correspond to the end of the lines and the local maxima to the beginning of them.
- **Y data:** the value of the data in each line it stays more or less constant when the writing is straight forming a descending step function which decreases in each line and ascends when there is a new page.

So, the first step is to find the position in which the page changes occurs. For that, both data are analyzed with the goal of avoiding false detections due to irregular writing. Analyzing the X data, as it was said before, it can be obtained the position of the candidates to be the start of the point. In Figure 3.3 (a) it can be observed the candidates to be the start of each line.

With the position of these candidates, it is time to analyze the Y data along the time. In this case, considering straight writing, the result is a slope function which decreases along the time as it can be seen in Figure 3.3 (b). In this Figure, it can be observed as well how the candidates to be a start of line coincides with the beginning of the slope and how the writing is not so straight. For that, it is computed the mean of the Y data of the samples in each bin resulted in the start line candidates. Analyzing the evolution of the curve formed by these points, the maximum value corresponds with the position of the new page.

If the data is dividing in the position given for this sample, the result is the document dividing into the different pages it has (Figure 3.4).

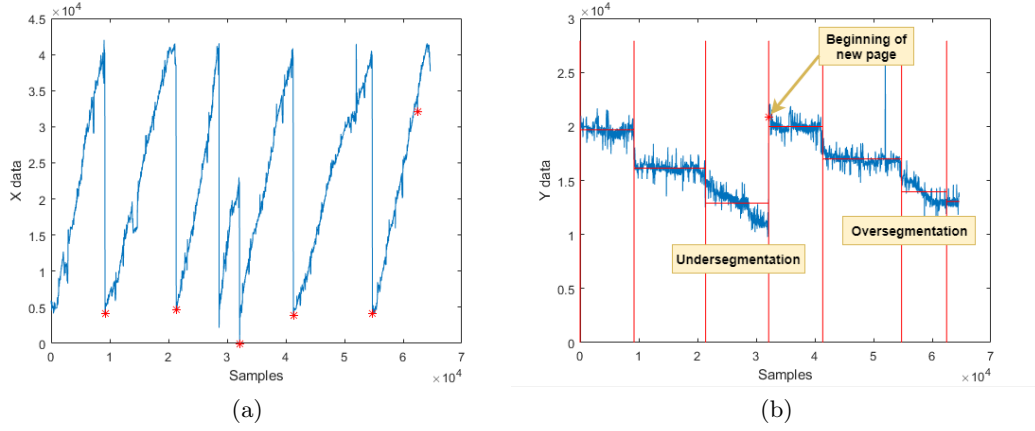


Figure 3.3: Evolution of the a) X data, b) Y data, along the time.

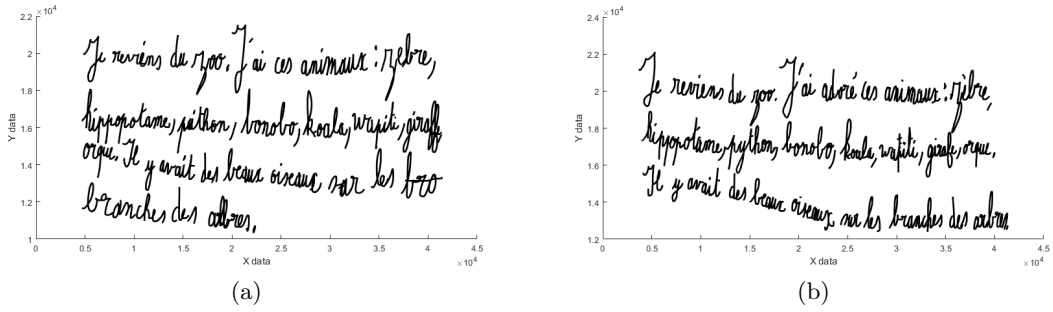


Figure 3.4: Pages obtained after the page segmentation procedure.

3.2 Line segmentation

Once the document is segmented in pages, the next step is to segment the pages in the different lines each one of them has. To do that, it is necessary another study of the X samples of each page. Searching for the local minima now, it was obtained the position of the start points of each line. As it can be observed in Figure 3.5, the number of local minima found is different than the obtained before, this is because the last line of the first page is too short compared with the rest and when analyzing the whole text this minimum is not considered and in the second page it existed an oversegmentation as it can be seen in Figure 3.3, that is why in this project it has been decided to carry out a more thorough analysis doing it in two steps.

As it was said, the samples where local minima are, correspond to the position of the new lines. In the example, dividing the data in the samples of the local minima it is obtained four different lines. Figure 3.6 presents the two first lines resulting of the segmentation.

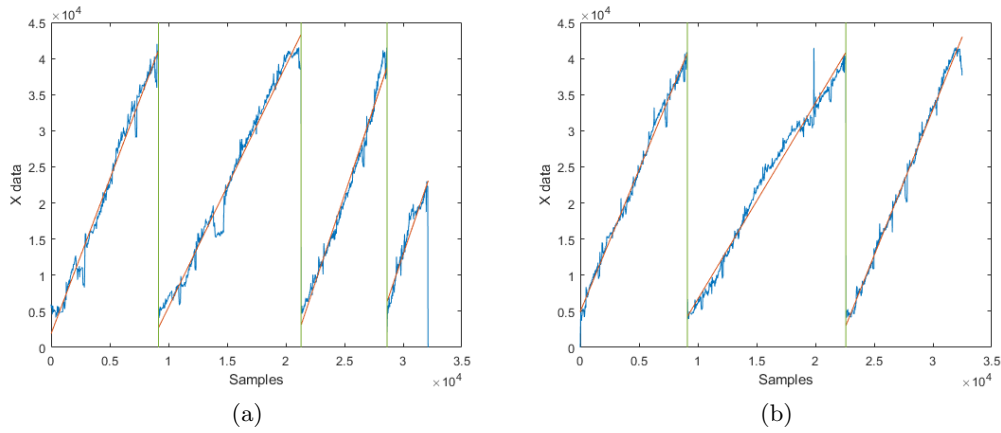


Figure 3.5: Evolution of X data of the a) first page, b) second page, along the time and the position of the beginning of the lines.

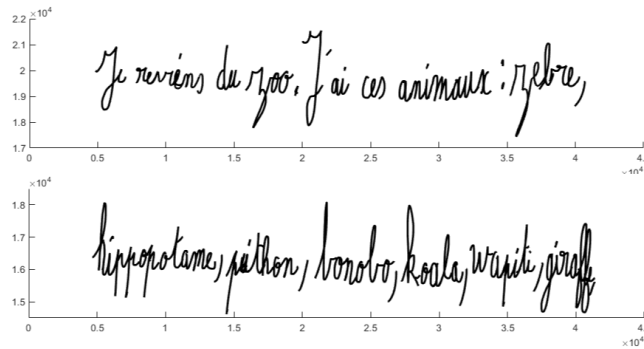


Figure 3.6: Two first lines of the first page.

3.3 Word segmentation

With the line segmentation fulfilled, it can be performed the segmentation of the lines into different words. To develop that, the first step is to divide the data of each line into strokes, that is, the trace between the “Pen-Down” and “Pen-Up” action when the user is writing down in the tablet. An example of these strokes are shown in Figure 3.7 when it can be observed how was built the words “du zoo”.

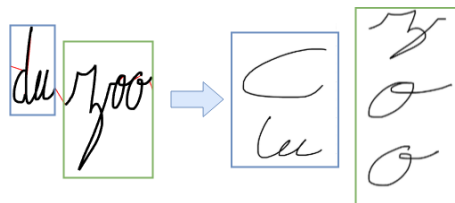


Figure 3.7: Strokes which form the phrase “du zoo”.

In the same figure appears in red the off-strokes that is, the union between two strokes.

The objective is to classify each one of the off-strokes obtained in each line in one of these classes: i) separation point (SP) and ii) non-separation point (NSP). In the SP class will be those off-strokes whose previous stroke and next do not belong to the same word (in the example the line between the “u” and the “z”).

To develop that, it is built a binary classifier which determines if an off-stroke is an SP and belongs to this class (1) or not (0). The classifier used is the Support Vector Machine (SVM). To train the classifier, it is necessary to extract the main features of the off-strokes and labeling each off-stroke as SP or as NSP. Figure 3.8 shows the outline of performing the classification using an SVM from a set of strokes obtained after, as mentioned, analyzing the “Pen-Down”, “Pen-Up” action of the pen on the tablet.

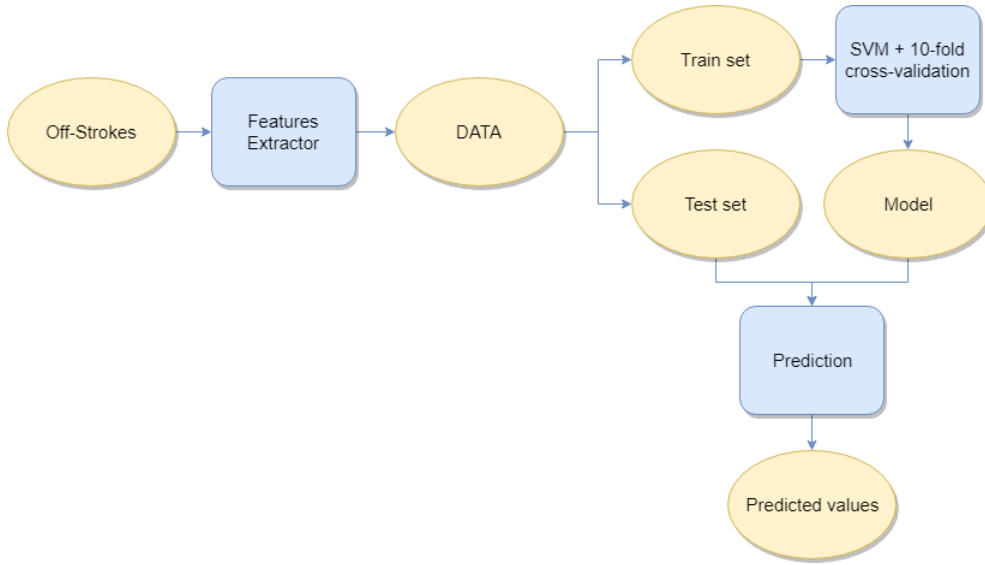


Figure 3.8: Diagram of the process of obtaining the prediction of the off-strokes based on a SVM. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.

3.3.1 Feature Extraction

The feature extraction consists in selecting a series of characteristics that a stroke has. Following [17] and [18] it was defined this terms which are represented in Figure 3.9:

- S_p : Immediate preceding stroke.
- S_s : Immediate succeeding stroke.
- B_p : Bounding box of S_p .
- B_s : Bounding box of S_s .
- B_{p_all} : Bounding box of all the preceding strokes.

- B_{s_all} : Bounding box of all the succeeding strokes.

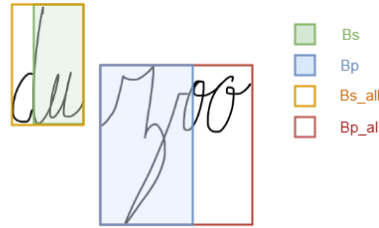


Figure 3.9: Terms used to define the strokes in the line “du zoo”.

With this, it was extracted the following features of the off-strokes:

- **F1**: Length of the off-stroke (e.g., Euclidean distance between the end of the S_s and the beginning of the S_p).
- **F2**: Angle of the off-stroke with respect to the X axis.
- **F3**: Distance between the centroids of B_{s_all} and B_{p_all} .
- **F4**: Overlap length between B_s and B_p in X.
- **F5**: Overlap length between B_s and B_p in Y.
- **F6**: Minimum point distance between S_s and S_p .
- **F7**: Angle between the vector from the centroids of B_s and B_p and X.
- **F8**: Ratio between B_s and B_p width.
- **F9**: Ratio between B_s and B_p height.

It is important to emphasize that, in order for the features to be comparable to each other, they must be normalized so that their values should be between zero and one.

3.3.2 Classification of features using SVM

To classify the off-strokes with its features mentioned above, it is used as binary classifier an SVM, as it was mentioned previously, which classifies the off-strokes in two classes, in this case, in the classes separation points or SP (1) and in the class, not separation point or NSP (0).

The Support Vector Machine or SVM is a learning algorithm which is related to problems of classification and regression. It is supervised because given a set of training labeled

samples, training the SVM with this set, it is built a model that predicts the class of a new sample. That is, SVM represents the sample points in space, separating the classes into two spaces as wide as possible by means of a separation hyperplane defined as the vector between the two points of the two classes closest to each other called support vector [21].

To build the model using SVM, it is necessary to define a kernel, these kernels functions are used to map the dataset into a higher dimensional space in order to make the data linearly separable. They can be linear, polynomial, radial basis function (RBF) or Gaussian. In this case, following [18], it was selected a radial basis function kernel.

Preparing the dataset

To build the classifier, firstly it is necessary to prepare the dataset splitting the data into train and test set. This partition normally is done using 70%-80% for the train set and the rest for the test. Figure 3.10 presents the division of the dataset, showing the way in which the data is organized. The rows represent in this case the different off-strokes and the columns its features. In order to perform a cross-validation afterward, the rows training data are shuffled. Finally, the last column corresponds to the label of the off-stroke, indicating if it is an SP (1) or not (0).



Figure 3.10: Representation of the dataset.

Preparing the validation set using k-fold cross-validation

In order to find the best set of features, it is carried out a k-fold cross-validation [22], with $k=10$, value widely used. With this method, it is used all the train data as training and as validation. The train set is divided into 10 folds, leaving one of them as the validation set and the rest as training data. The process of cross-validation is repeated ten times with the aim of each fold has been the validation set. Figure 3.11 shows the process of k-fold cross-validation in the case of $k=4$.

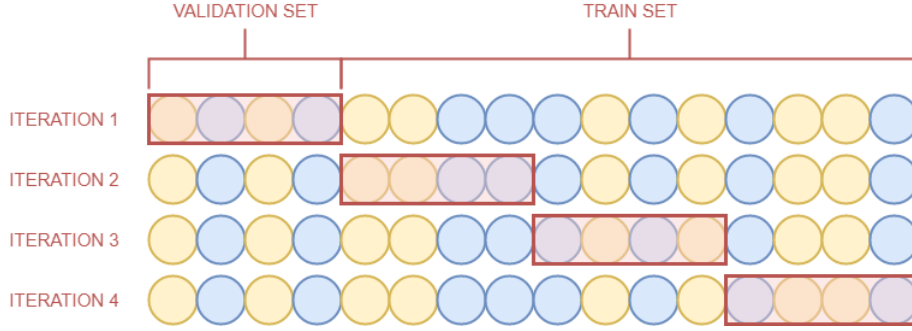


Figure 3.11: Division of data when it is used 4-fold cross validation.

The goal of using the cross-validation here is that, given the number of features that will be used to train the model, different combinations of features are used to train the model in each iteration to get the best set of features. The best set of features will be the one that minimizes the prediction error with respect to the validation data.

Feature selection

As mentioned before, to obtain the best set of features it is necessary to indicate the number of features to use. To know what is the optimal number of features to train the model, the following steps have been taken one hundred times:

1. Shuffle the rows of the train set.
2. Divide the train set again into another train set and test set in proportions 70%-30% respectively.
3. Perform the 10-fold cross-validation for a different number of features (in this case, from only one feature to nine features) to obtain the best combination of features.
4. With the best combination of features using a different number of features, train the model and measure its performance using the new test set. In the metrics TP and TN are the number of true positives and true negatives, respectively, while FP and FN indicate the number of false positives and false negatives, respectively:

- **Accuracy:** it is the percentage of correct predictions.

$$Accuracy = \frac{TP + TN}{Total} \quad (3.1)$$

- **Recall:** it is the percentage of positive samples that have been detected correctly.

$$Recall = \frac{TP}{TP + FN} \quad (3.2)$$

- **Precision:** it is the percentage of detections made that are correct. It is a measure of reliability of the detections.

$$Precision = \frac{TP}{TP + FP} \quad (3.3)$$

- **F1-score:** harmonic mean between precision and recall.

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision} \quad (3.4)$$

At the end of the 100 iterations, it is computed the average of the metrics of the results for each number of features chosen. The result obtained is the graph presents in Figure 3.12 where it is shown the evolution of these metrics when it is used more or fewer features. As it can be observed, the performance increases when using the nine features defined, however, the precision is higher using only four features so, with four features it can be achieved a good model.

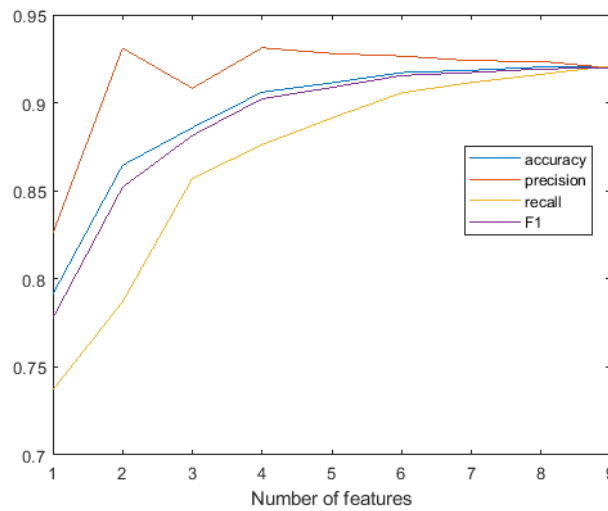


Figure 3.12: Average of the performance metrics of the SVM after executing it 100 times for different number of features.

It should be noted that this step is only done once with the aim of knowing how many features are necessary for good training of the model, once the study is done, the number of features that provide good results is chosen.

Testing the model

Once the model is created selecting the number of features and the features are going to be used, it is carried out the prediction with the test set it was left out in the train set (see Figure 3.10). The prediction of the model consists, for each data, in a score and a

label where, depending on how high or low the score is, the test data is labeled as SP or NSP.

There are several ways to test the performance of the model, for example, the metrics used before to select the number of features. Another way to indicate the performance of the model is using the confusion matrix which indicates the number of TP, TN, FP and FN that it was obtained in the prediction. However, the result of these metrics and the conclusions are shown in Chapter 6.

Chapter 4

Warping letters in a word overview

In this chapter, it is going to be explained the method proposed by C. Sansalvadó [1] on the basis of which this thesis is developed. In that project, using different templates, the position of the letters of a word presented in an image is determined in order to segment the word into the letters that form it. To perform that, it is combined template matching, specifically using the Lucas-Kanade algorithm, with graph theory, as it can be seen in the scheme presented in Figure 4.1, in order to find the optimal solution which is the one that finds the greatest similarity between handwritten letters and templates. In this procedure it was assumed that the words are known *a priori* which helps to match each letter with its corresponding template without the need for character recognition. In section 4.1 it is explained the first part of the algorithm, corresponding to the application of Lucas-Kanade to warp the templates and section 4.2 details the second part, corresponding to the graph-based solution. In the Chapter 5 it is explained the word segmentation process based on the method explained here.

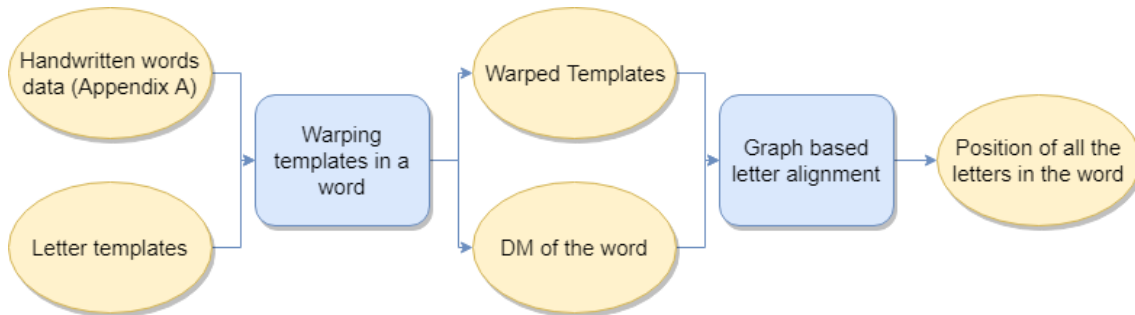


Figure 4.1: General scheme of the method for warping letters in a word. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.

4.1 Lucas-Kanade algorithm

To obtain the warped templates used to compute the optimal position of the letters in a word, as mentioned before, it is used a template matching technique based on image alignment which consists of moving and deforming a template to minimize the difference between the template and the image. For this purpose, it is used Lucas-Kanade algorithm [23] which is an optimization method for template matching under the sum of squares differences (SSD). The goal of this algorithm is to minimize the SSD error between two images when align a template image $T(\mathbf{u})$ to an input image $I(\mathbf{u})$, where $\mathbf{u}=(u,v)^T$ is a column vector containing the pixel coordinates. The parameterized set of warps, $\mathbf{W}(\mathbf{u};\mathbf{p})$, where $\mathbf{p}$ is the vector which contains the parameters for the warping, takes the set of pixels $\mathbf{u}$ of the template T and maps it to the sub-pixel location $\mathbf{W}(\mathbf{u};\mathbf{p})$ of image I . So, the problem consists in minimizing this function of error using the iterative LK algorithm [23] in which the set the parameters $\mathbf{p}$, in this case, it is only considered the translation and the scaling, is updated in every step until the convergence:

$$error(\mathbf{u}) = \sum_{\mathbf{u}} [I(\mathbf{W}(\mathbf{u};\mathbf{p})) - T(\mathbf{u})]^2 \quad (4.1)$$

To achieve the goal of Lucas-Kanade algorithm, it is necessary to perform several steps before which are presented in Figure 4.2 where it is shown the process to obtain the warped templates from the input data collected by the tablet and from the letter templates.

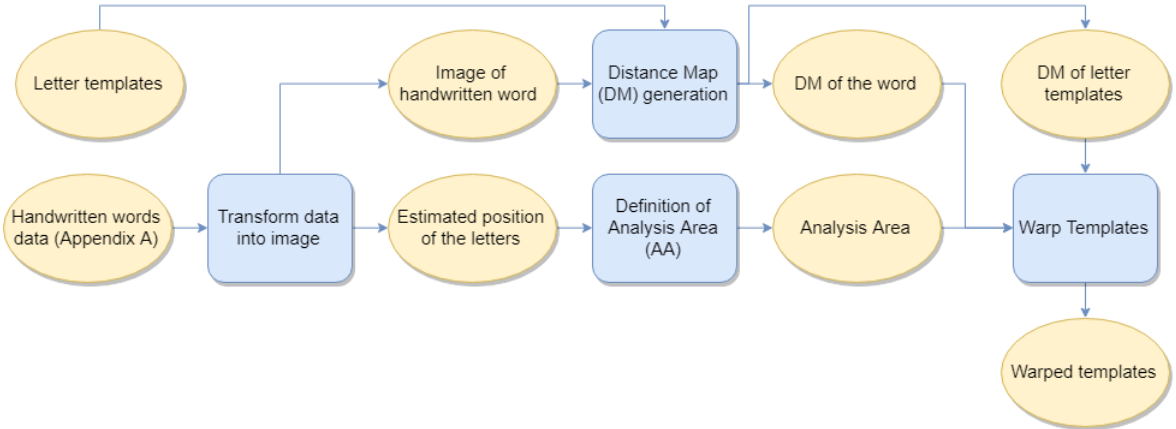


Figure 4.2: Outline of the process of obtaining the templates. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.

4.1.1 Image and distance map generation

The first step to develop the template warping is the pre-processing of the given image. The data used, as it is explained in the Appendix A, was recording by a tablet and stored

as a set of points along the time. This set of points is transformed into an image in order to process it and to take some useful information from it.

As the word which is being analyzed is known *a priori*, the number of letters that it has is known as well. So, the word is divided into the number of letters that it has, each one with the same width assuming a regular distribution of the text, taking into account the height of each letter (Appendix A). With this criterion, the estimated position of each letter is obtained through a bounding box. Figure 4.3 presents the estimated position of the letters of the word “glace”.

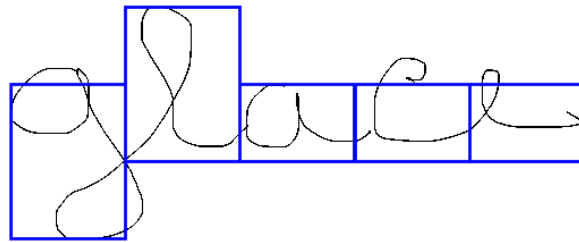


Figure 4.3: Estimated positions of the letters of the word “glace”.

The way to analyze the correlation between image and template consists in computing the distance map, hereinafter *DM*, both the whole word and of each one of the templates corresponding to each letter of the word. The *DM* is generated by labeling all background pixels with the value of the distance of their closest non-background pixel (i.e., word contour) which allows seeing how strongly linked is the result to the distance metrics used for computation, in this case, the Euclidean distance. To carry out the *DM*, it is necessary to convert the image into gray-scale, invert the image, perform a dilation with the goal of obtaining a more robust contour and then binarize the image. The reason for using this technique is to improve the computation cost and to avoid abrupt changes in the image function. The *DMs* of the reference templates of each letter are also calculated. In Figure 4.4 it can be seen an example of a *DM* of the word “glace”.

4.1.2 Analysis area definition

To limit the number of positions than a warped template can have on an image, it was defined an Analysis Area, AA. This area is the region of the image in which the LK algorithm is applied. This area consists in the area of each bounding box (the estimated position of the letters in the image of the word) created before (see Figure 4.3) plus a fixed magnitude added along the two dimensions of the template, m_u and m_v for the u and the v coordinates of the template respectively, as it is presented in Figure 4.5. These magnitudes are proportional to the original letter width which is used as a reference.

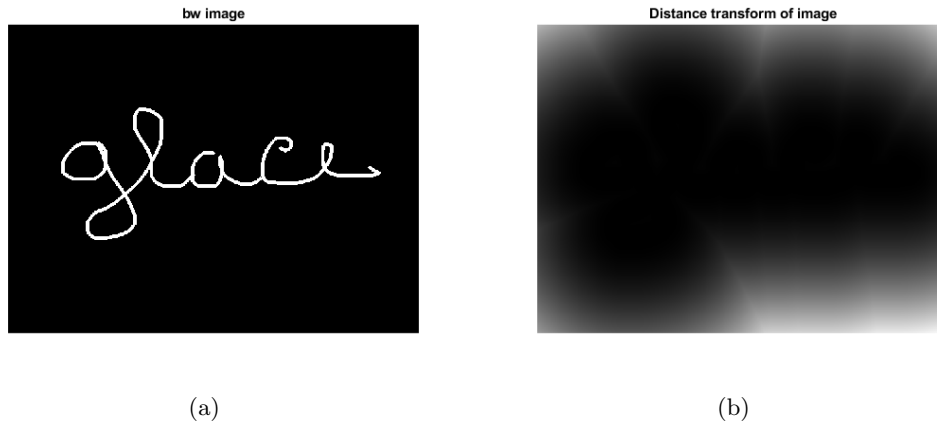


Figure 4.4: Distance map of the written word “glace”.

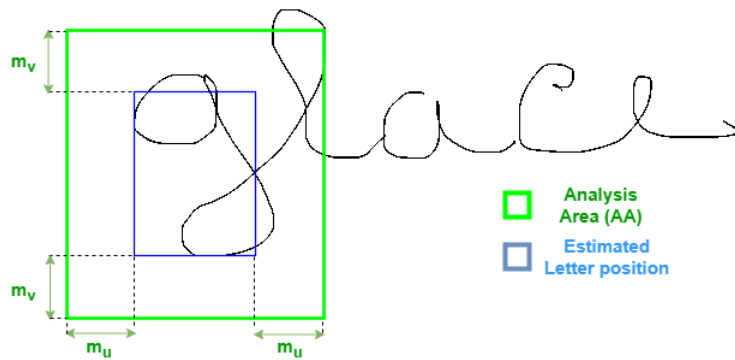


Figure 4.5: Estimated position and AA of letter “g” in the word “glace”.

4.1.3 Template warping using Lucas-Kanade algorithm

As it was said before, Lucas Kanade finds the best set of warping parameters of the template, $\mathbf{p}$, which minimize the error function between the warped template and the image (equation 4.1) in the AA described previously. To perform this, it is used both, the *DM* of the whole word and the *DM* of each template [23].

In this case, as it was said before, the set of parameters $\mathbf{p}$ contains translation and scaling parameters. Thus, the result it is defined by the size and the position of the warped template within the image I , which in this case, it has to be borne in mind that the image I is that which is within the AA. In the case of the word “glace”, the result of the best template matching of the letter “g” and its warped template are shown in Figure 4.6.

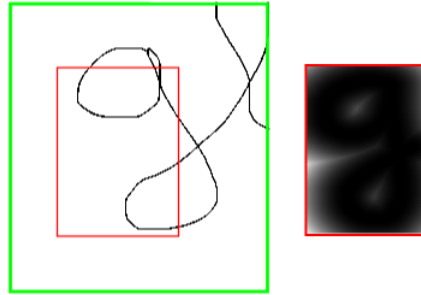


Figure 4.6: Best template matching result using LK algorithm and its corresponding warped template.

4.2 Graph based letter alignment

The position obtained using Lucas Kanade algorithm corresponds with the best match between the template and the image only in the area limited by the AA, but it does not have to coincide with the best matching considering the complete word. To obtain the best position of each template in the word it is necessary to perform an additional translation of each template considering (i) the correlation between the displaced warped template and the word and (ii) the distance between two consecutive warped templates. The way of performing that consists of building a directed graph in order to obtain the optimal combination of the positions of the templates using a shortest-path algorithm. Figure 4.7 presents the diagram of blocks of this procedure.

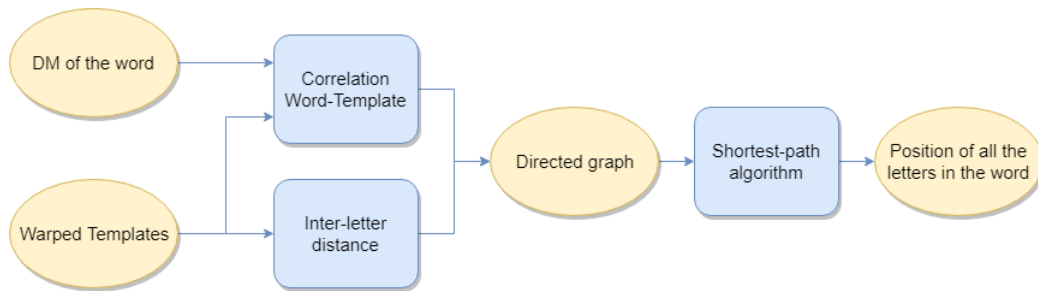


Figure 4.7: Outline of the graph based letter alignment method. Notation: the oval blocks indicate input or output data and the rectangular blocks indicate processes.

4.2.1 Building the directed graph

As it was said before, to obtain the best combination of the position of templates it is built a graph which contains the correlation between the word and the template in each possible position and the distance between two templates. Taking this into account, to build the graph it was established that the edges of the graph contain the distance between templates and the nodes the cost given by the correlation between the template and the

word in a specific position.

The possible positions of the templates are given by a new AA, created similarly to the view in Figure 4.5 but using the template position calculated by LK instead of the estimated position, and the search window, SW, in which the templates can be move, taking into account that the reference point of the templates are their upper-left corner. Figure 4.8 shows the SW.

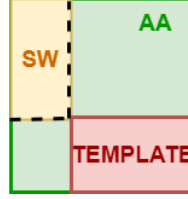


Figure 4.8: Representation of the search window, SW.

Correlation word-template

The correlation between the word and the template is given by the Normalized Cross Correlation, NCC [24] which takes into account the image of the *DM* of the word in the AA and the template. The results are in the range $[-1,1]$ where the higher the value is, the higher correlation between the image and the template.

To give the cost associated to the value given by the NCC, it was proposed a cost function which penalizes the results which are not close to the maxima as it can be seen in Figure 4.10 (a).

Inter-letter distance

The inter-letter distance between the k -th template of the word and its immediate consecutive is defined as the Euclidean distance between the point located more to the right of the k -th letter and the point located more to the left of the $(k+1)$ -th letter. Figure 4.9 shows these points in the template of the letter “o”.

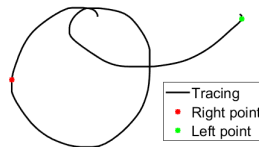


Figure 4.9: Left and right points of the template of the letter “o”.

This inter-letter distance is defined as follows and it is important to consider if the

right point of the k -th letter is placed before the left point of the $(k+1)$ -th letter or after and in the next formula this aspect appears as the sign:

$$d(P_{right_k}, P_{left_{k+1}}) = \text{sign}(u_{right_k} - u_{left_{k+1}}) \sqrt{(u_{right_k} - u_{left_{k+1}})^2 + (v_{right_k} - v_{left_{k+1}})^2} \quad (4.2)$$

As in NCC, a cost function associated with the inter-letter distance is defined as the function presented in Figure 4.10 (b), where the cost increased when the absolute value of the ratio between the distance and the letter width increases.

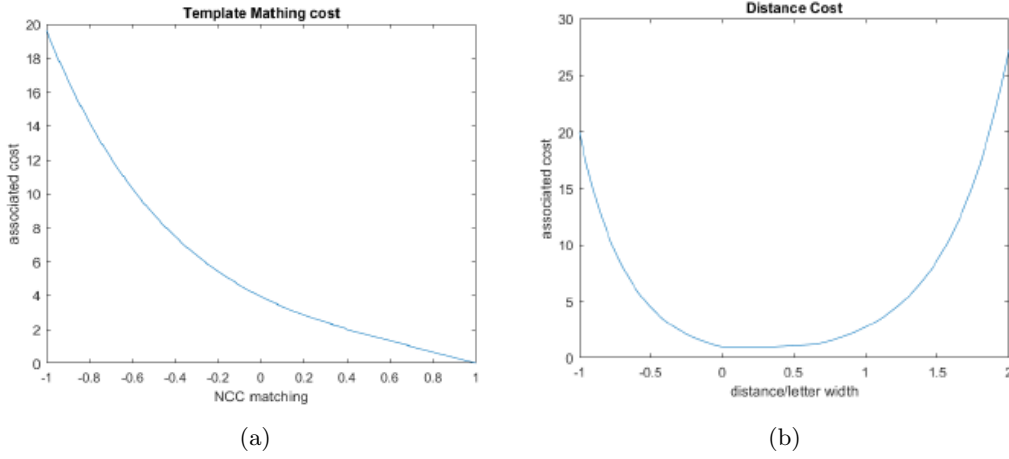


Figure 4.10: Cost function of: a) NCC and b) inter-letter distance. Image from [1].

Graph representation

With the cost of placing the warped template in each pixel of its SW and the cost given by the distance between two consecutive templates it can be built a directed graph to apply later a shortest path algorithm in which the nodes of the graph correspond to the cost of the translation of the warp template, NCC cost, and the edges to the distance between two consecutive templates in a word. Figure 4.11 shows this graph.

At the beginning of the graph there is a node which is connect to all the possible positions of the SW of the first letter. The edge connections $D_{i,m-j,n}$ represent the distance between two consecutive templates, the i -th and the j -th, when the first one is placed in the position m of its SW and the second one is situated in the position n of its SW. $P_{i,m}$ indicates the NCC cost of placing the i -th template in the position m of its SW.

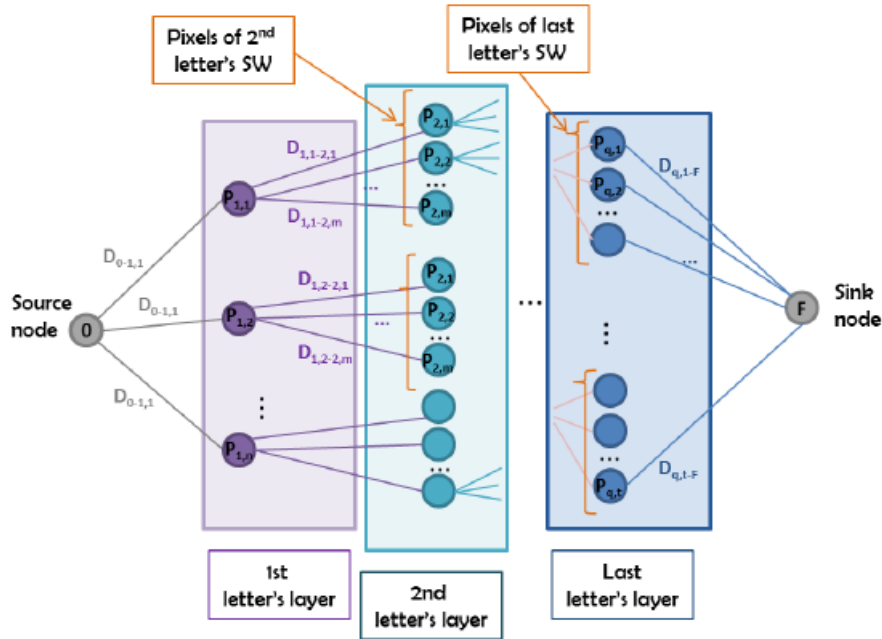


Figure 4.11: Graph representation for finding the best positions of the letters in the word. Image from [1].

4.3 Results of the method

The result after applying the shortest path algorithm (in this case, Dijkstra [25]) is showed in Figure 4.12, where is compared with the previous position of the bounding boxes used before.

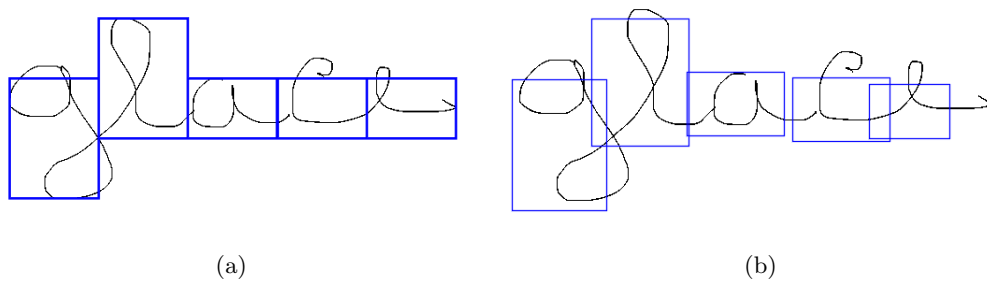


Figure 4.12: Comparison between: a) the estimated position of the letters and b) the position of the letters obtained after applying LK algorithm and the graph-based letter alignment.

With the final position of the letters in the word, it can be computed the endpoints of each letter in order to segment the word in the different letters that it has, as it is explained in the next chapter, Chapter 5, using two different techniques.

Chapter 5

Letter segmentation

In this chapter, it is going to be explained how to perform the segmentation of words into letters based on the use of Lucas-Kanade template warping carried out by C. Sansalvadó [1] and explained in Chapter 4. The goal is, based on the positions of the templates given by the Lucas-Kanade algorithm and the graph-based letter alignment, to obtain the endpoints of the letters which form the whole word in order to segment it.

The proposed method consists of measuring the distance between each pair of points of the function of time which defines the tracing of the word recorded by the tablet and the templates corresponding to each letter. With this, an estimation of the final positions of each of the letters will be made, as it will be seen in the following sections. Figure 5.1 presents the outline of this chapter, indicating with different colors the domains used because to obtain the endpoints it is necessary to consider the function of time recording by the tablet, as it will be presented in this chapter.

To check the effectiveness of the proposed method, the results obtained will be compared with those provided by the Ground Truth, which has been manually defined, as it will be seen later in the Chapter 6.

5.1 Data preparation

First, it is important to clarify that, in order to calculate the distance between the curve $g(t)$ corresponding to the trajectory of the pen on the tablet along the time, where t denotes the samples of the curve sorted in time, and the templates, everything has to be in the same domain and has to be comparable, that is, the image of the word generated to apply the LK algorithm (Chapter 4) and the curve $g(t)$ must match. To do that, the curve $g(t)$ has to be adapted to the proportions and the position of the word image, which is in other domain, the domain of the templates. As seen in Figure 5.2, the size and the position of the images of the word “glace” is not the same. Therefore, as will be seen below, a scaling and translation transformation of the curve $g(t)$ will be performed.

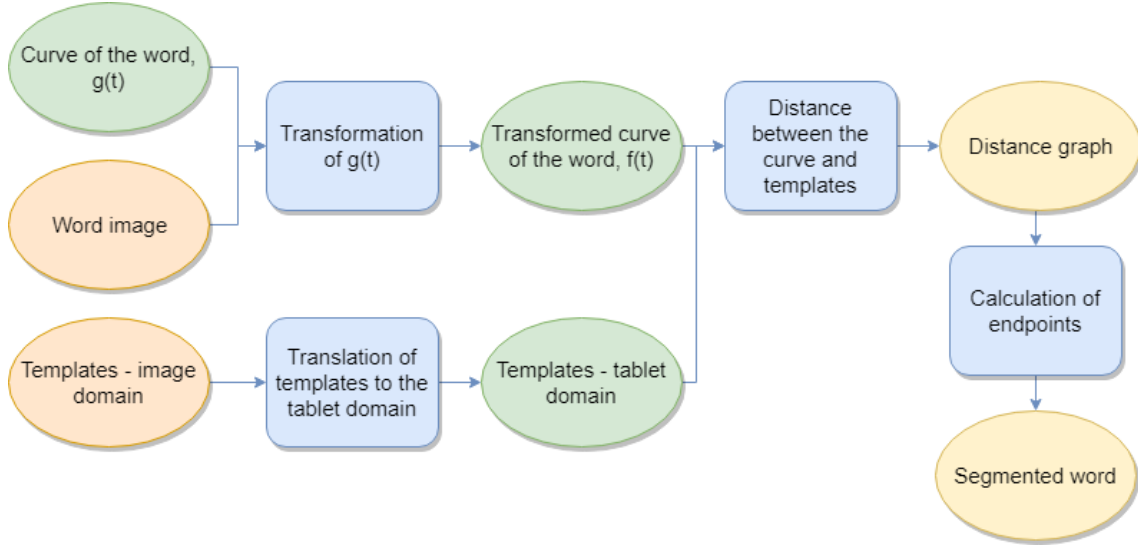


Figure 5.1: General scheme of letter segmentation. Notation: the oval blocks indicate input or output data, of these, the greens indicate that the data is in the coordinate system of the tablet and the oranges in the image domain, and the rectangular blocks indicate processes.

To do that, it was defined two different coordinate systems. Figure 5.3 shows in the left the coordinate system of the image domain, (U,V) , and in the right the tablet coordinate system, (X,Y) .

Transformation of the curve $g(t)$ into the curve $f(t)$

As previously mentioned and as seen in Figure 5.2, it has to be done the images comparable in order to be able to measure the distance between the curve $g(t)$ and the templates. Comparing both images, it can be seen that there are a scaling and a translation of one with respect to the other. The objective is to obtain these transformation parameters: two corresponding to vertical and horizontal scaling and a translation along both axes. Thus, the parameters that are needed are four: two correspondings to the scale factor and two corresponding to the translation.

Mathematically, the transformation of a point of the curve $g(t)$, $P_g=(x_g,y_g)$, to placed the curve in the same position of the image of the word obtaining the transformed point, $P_f=(x_f,y_f)$, can be written as follows. At this point, it is considered that the Y axis and the V axis have the same direction:

$$x_f = r_u + x_g * s_u \quad (5.1)$$

$$y_f = r_v + y_g * s_v \quad (5.2)$$

In the previous equations, r_u and r_v represent the translation of the point horizontally

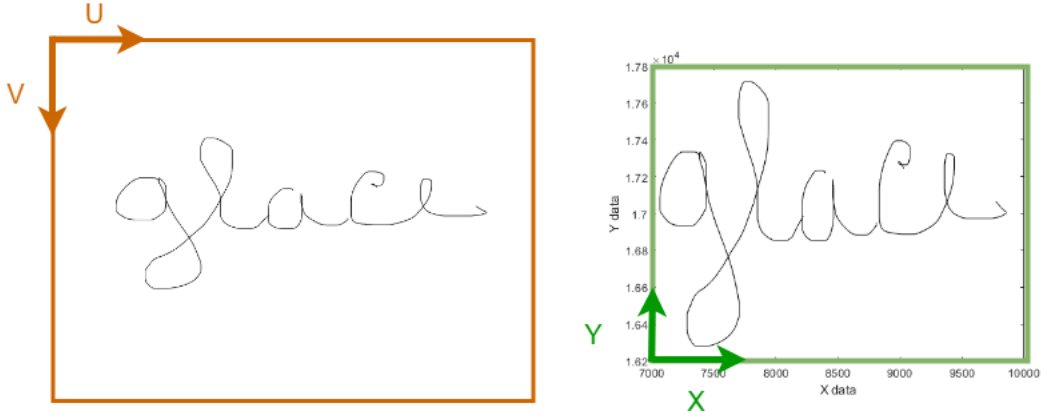


Figure 5.2: Representation of the word “glace” in left) image system and right) tablet domain.

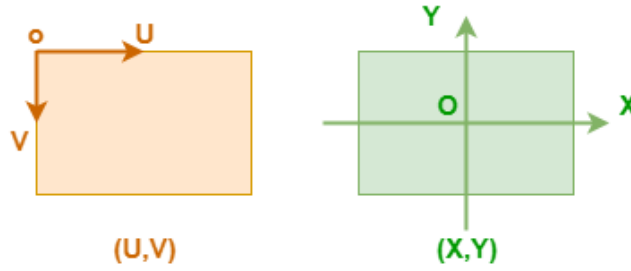


Figure 5.3: Systems of coordinates used. Left: Coordinates of a template system (U,V). Right: tablet coordinates system, (X,Y).

and vertically, respectively, while s_u and s_v represent the scale factor of width and height respectively. In Figure 5.4, it is shown the parameters which define the image in each of the coordinate systems used. With this in mind, the four previously defined parameters that are needed to perform the transformation can be defined as follows:

$$r_u = \Delta u - \Delta x * s_u \quad (5.3)$$

$$r_v = \Delta v - \Delta y * s_v \quad (5.4)$$

$$s_u = \frac{w}{W} \quad (5.5)$$

$$s_v = \frac{h}{H} \quad (5.6)$$

In this way, the transformed curve is $f(t)$ resulting of transforming each (x_g, y_g) point of the curve into (x_f, y_f) applying the equations 5.1 and 5.2 respectively.

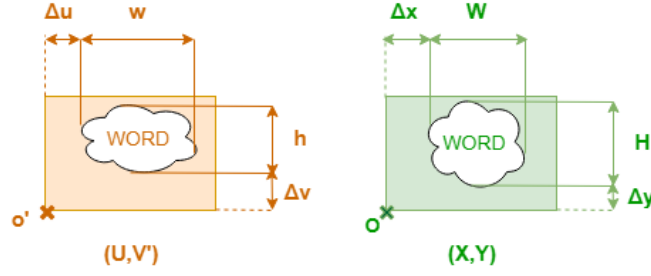


Figure 5.4: Parameters which define the image in the template domain (left) and in the tablet coordinates (right).

Transforming the templates to the tablet domain

With the new curve $f(t)$, a point from the image of the image and a point from the curve can be compared. However, as it can be observed in Figure 5.3, the direction of the Y axis and the direction of the V axis are not the same. To simplify the calculations, it is used the tablet system. So, the v coordinate of the points of the image of the word has to be re-calculated. To do that, the diagram showed in Figure 5.5 is going to be considered. In this image, the parameters needed to perform this change are shown. The yellow region represents the area of interest or ROI that is where the image of the word is located. Drawing an imaginary line horizontally in the middle of the ROI, it can be observed that the distance of the current vertical coordinate and the desired one with the horizontal line, δy , is the same, that is, the line acts as a mirror on the Y axis. Taking this into account, the transformation of the v coordinate of a point $Q'=(u,v)=(x,v)$ to obtain the point $Q=(x,y)$ is:

If v is below the horizontal line (as in Figure 5.5):

$$y = \Delta y + \frac{h}{2} + \delta y = \Delta y + \frac{h}{2} + (\Delta y + \frac{h}{2} - v) = 2 * (\Delta y + \frac{h}{2}) - v \quad (5.7)$$

If v is above the horizontal line:

$$y = \Delta y + \frac{h}{2} - \delta y = \Delta y + \frac{h}{2} - (v - (\Delta y + \frac{h}{2})) = 2 * (\Delta y + \frac{h}{2}) - v \quad (5.8)$$

5.2 Distance between data and templates

Once it is computed the new curve, $f(t)$, and the vertical coordinate change of a point of the image in order to make the image of the word and the curve comparable, this section will present how to calculate the distance between the curve and each of the warped templates that define the word. For the k -th template corresponding to the k -th letter, the distance between $f(t)$ and its corresponding template, T_k , can be written as follows:

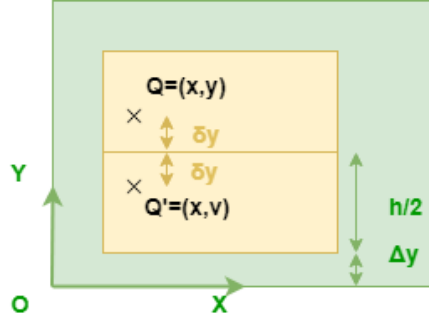


Figure 5.5: Reference parameters to take into account to make the value change of the Y axis.

$$\xi_k(t) = d(f(t), T_k) \quad (5.9)$$

To define the distance defined above, it has been developed two different methods. The first one consists in using the center of each template and the second one involves the distance map of the template. The goal is to obtain a set of endpoints which are the timestamps in which each letter of the word ends.

5.2.1 Using the center of the template as reference

Since the template is defined by the coordinates of the left upper corner, its width and its height, it is easy to determine the center of the template, CT . With the k different CT s which define the whole word in the tablet domain, the distance between the k -th CT and the function which describes the word, $f(t)$, can be computed using the Euclidean distance for each point of the curve as follows:

$$\xi_k(t) = d(f(t), CT_k) = \sqrt{(x - x_{CT_k})^2 + (y - y_{CT_k})^2} \quad (5.10)$$

In the Figure 5.6 it can be shown the tracing of the word “glace”, the position of the templates computed before using Lucas-Kanade template warping explained in Chapter 4 and the center of each of the templates.

The result obtained when calculating the distance of each point to the center of each template is shown in Figure 5.7. The important thing about this graph are those points of the tracing whose distance to the center of a template is minimal since it means that those samples are closer to that template than others, that is, to the letter corresponding to that center. In the lower part of the graph, where the curve-template distances are minimal, it can be observed as in the first samples the minimum distance corresponds to the letter “g”, then to the letter “l”, thus until the last letter, in this case the letter “e”. It is important to take this into account in order to perform the segmentation of words

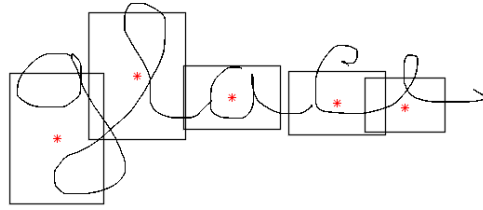


Figure 5.6: Tracing of the word “glace”, the position of the templates and their corresponding centers.

into letters, as it will be seen in section 5.3.

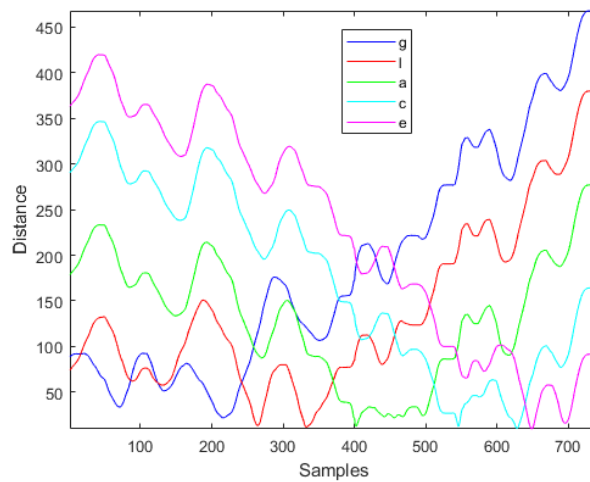


Figure 5.7: Distance graph between the center of templates of the word “glace” and each point of the tracing.

5.2.2 Using the distance map of the template as reference

The other procedure to calculate the curve-template distance is to use the distance map of each template, DM , instead of using the center of it. The distance map, as explained in section 4.1.1, is a grayscale image associated to a binary image (in this case, to the tracing of a letter) in which the value of each pixel corresponds to the minimum distance to an obstacle pixel in the binary image. So, in the pixels where the letter tracing is, the distance will be minimal and is represented with a tone closer to black as can be seen in Figure 5.8.

The idea is to put the distance map above the tracing of the word, as it can be observed in Figure 5.9, in its corresponding position (in the tablet domain) and look for each point of $f(t)$ what is the corresponding value in the DM . If the points do not have an associated distance map value either because there is no template on them (for example, at the top

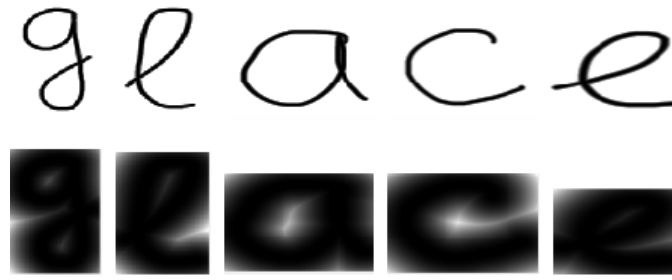


Figure 5.8: Top: Letter templates of word “glace”. Bottom: Distance map of the templates of word “glace” after applying Lucas Kanade.

of letter “c” in Figure 5.6) or because the point is far of the current template (for instance, the distance of a point of “g” to the template of “e”), the distance to the nearest pixel of the template is calculated as it is explained below.

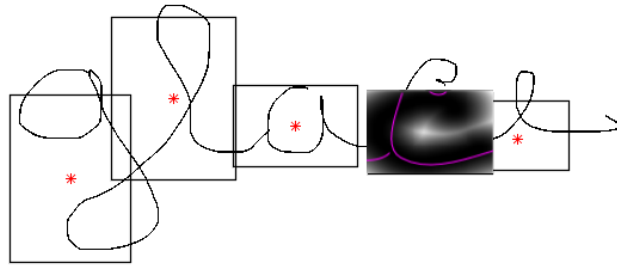


Figure 5.9: Distance map of the letter “c” above the tracing of the word.

Global and local coordinates

Global coordinates are those that include the complete word, that is, what is shown in the right picture of Figure 5.3 and local coordinates are the ones that each *DM* has, as it can be observed in Figure 5.10.

The result of applying LK algorithm with the graph-based letter alignment gave the position of the templates in the global coordinates but what it is desired is, given a point of the curve $f(t)$ defined as (x,y) , get the value of this point in the local system of each template in order to obtain the value of the DM corresponding to this point. For this it is defined the change between local and global coordinates in the following way (taking into account that it has already been made for each DM the change to the coordinates of the tablet, that is, the vertical axis has been changed):

$$x' = x - \Delta x \quad (5.11)$$

$$y' = y - \Delta y \quad (5.12)$$

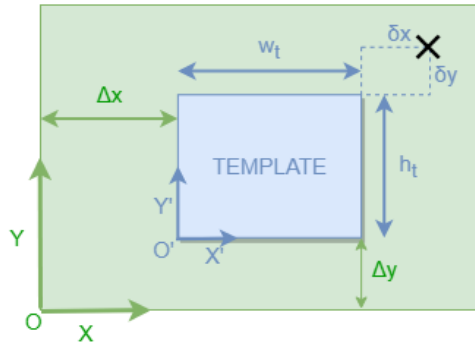
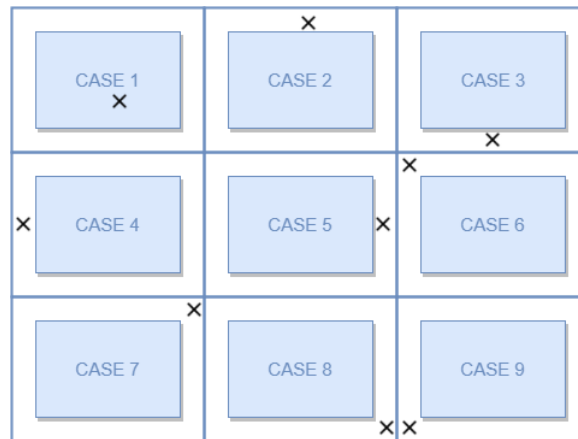


Figure 5.10: Global and local coordinates of a template.

Depending on where a point of the curve $f(t)$ is placed with respect to the DM , the distance is computed differently following the graph presented in Figure 5.11 where 9 cases are identified.

Figure 5.11: Different cases of positions that the points of the curve $f(t)$ can have with respect to the DM (represented as a blue rectangle).

Formally, the distance in these cases from the function $f(t)$, which describes the word written, and the distance map of the k -th letter, DM_k , can be written as follows:

$$\xi_k(t) = \begin{cases} DM_k(x', y') & \text{if case 1} \\ DM_k(x', h_{t_k}) + \delta y & \text{if case 2} \\ DM_k(x', 0) + \delta y & \text{if case 3} \\ DM_k(0, y') + \delta x & \text{if case 4} \\ DM_k(w_{t_k}, y') + \delta x & \text{if case 5} \\ DM_k(0, h_{t_k}) + \sqrt{\delta x^2 + \delta y^2} & \text{if case 6} \\ DM_k(w_{t_k}, h_{t_k}) + \sqrt{\delta x^2 + \delta y^2} & \text{if case 7} \\ DM_k(w_{t_k}, 0) + \sqrt{\delta x^2 + \delta y^2} & \text{if case 8} \\ DM_k(0, 0) + \sqrt{\delta x^2 + \delta y^2} & \text{if case 9} \end{cases} \quad (5.13)$$

The result obtained is similar to the obtained using the center of the templates and it has the same properties, as it can be shown in Figure 5.12.

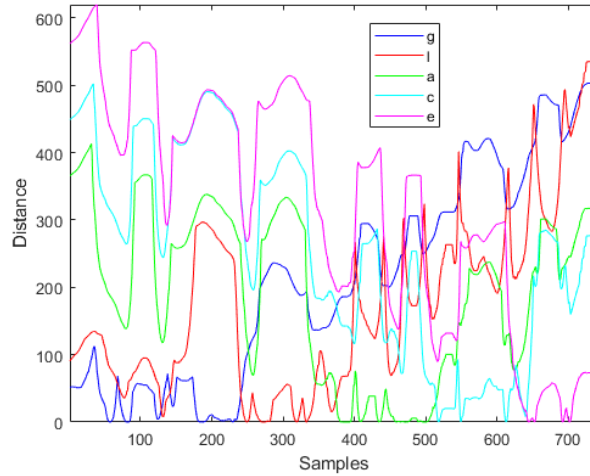


Figure 5.12: Distance graph between the distance map of templates of the word “glace” and each point of the drawn.

5.3 Calculation of the endpoints of the letters

In this section is going to be explained how the endpoints, e , of the letters are determined. The endpoints indicate the timestamp of the curve $f(t)$ in which a letter ends. To calculate them it is necessary a distance like the one in the Figure 5.7 and the one in Figure 5.12. As mentioned above, the important points for each template are those whose distance is minimal since it indicates that they are closer to each letter and therefore, that they can belong to that letter. Bearing in mind that a point can belong to a letter when its

distance to that letter is minimal, if distance is defined as cost, the cost of a point to its corresponding template is minimal and therefore the sum of the total cost of all points is minimal. So the problem becomes a minimization problem where the optimal combination of the endpoints of letters that minimizes the cost, $\mathbf{e}$, has to be found. Formally, for a word with N letters, the total cost can be written as:

$$D(\mathbf{e}) = D(e_0, e_1, \dots, e_N) = \sum_{k=1}^N \left[\sum_{j=e_{k-1}}^{e_k} \xi_k(j) \right] \quad (5.14)$$

Looking at the equation above, it exists an endpoint for $k=0$, e_0 , this point indicates the beginning of the word in the curve $f(t)$, that is $f(t=0)$. So, the optimal cost when starting from the initial state e_0 is:

$$D^*(e_0) = \min_{e_1, \dots, e_N} \left\{ \sum_{k=1}^N \left[\sum_{j=e_{k-1}}^{e_k} \xi_k(j) \right] \right\} \quad (5.15)$$

To find the set of endpoints several restrictions have to be defined to help solve the minimization problem.

- The endpoint e_0 when $k=1$ is corresponding to the first sample, that is, $e_0=0$. Likewise, the endpoint e_N is corresponding to the last sample $e_N=S$, where S is the number of samples.
- The endpoints cannot be negative since the timestamp is always positive.
- The endpoint of the k -th letter, e_k (for $k=1, \dots, N-1$), cannot be lower than the endpoint of the previous letter, e_{k-1} , nor higher than the endpoint of the next letter, e_{k+1} since if so, the correct order of the letters of the word would be changed.

Taking this into account, the complete minimization problem can be written in the following way:

$$\begin{aligned} & \underset{e_1, \dots, e_N}{\text{minimize}} && \sum_{k=1}^N \left[\sum_{j=e_{k-1}}^{e_k} \xi_k(j) \right], \quad k = 1, \dots, N. \\ & \text{subject to} && e_0 = 0 \\ & && e_N = S \\ & && e_{k-1} < e_k < e_{k+1}, \quad k = 1, \dots, N-1. \\ & && e_k > 0, \quad k = 1, \dots, N. \end{aligned} \quad (5.16)$$

5.3.1 Solving the cost function

The first step to solve the minimization problem presented above consists in looking for the candidates to be an endpoint in each level, considering that a word with N letters has $N+1$ levels (starting at 0). The candidates to be an endpoint of level k are the intersection points between the curves $\xi_k(t)$ and $\xi_{k+1}(t)$, for $k=1, \dots, N-1$ that satisfy the restrictions of the minimization problem.

Once the set of candidates has been found, it can be defined a graph in order to find the set of endpoints, $\mathbf{e}$, which minimizes the cost function using a shortest path algorithm as Dijkstra. To define this graph, it has been considered:

- Set of $N+1$ set of candidates. The first set contains only the initial endpoint, e_0 , what, as described previously, is the first point of the tracing of the word. Likewise, the last set only contains the final endpoint, e_N , which is the endpoint of the word.
- Candidates to be an endpoint. The k -th ($k=1, \dots, N-1$) contains M candidates, each one of them are denoted as $e_{k,m}$, where k is the number of the set and $m=1, \dots, M$ is the number of the candidate in this set.
- Each edge between two states or nodes has an associate cost which is the distance showed in the distance graph:

$$D_{(k-1,l)-(k,m)} = \sum_{j=e_{k-1,l}}^{e_{k,m}} \xi_k(j) \quad (5.17)$$

This equation indicates the distance between the m -th endpoint of the k -th set and the l -th endpoint of the previous set.

In the Figure 5.13 it can be seen the diagram of the shortest path algorithm taking into account the considerations mentioned above.

As mentioned previously, one way to solve the shortest path problem is using Dijkstra's algorithm [25]. This algorithm calculates the shortest path of a set of nodes given the initial and the final ones and the cost of the edges between nodes. It has a complexity of $O(|E|^2)$, where E is the number of nodes present in the graph. The relationships between nodes can be represented in a matrix form, called cost matrix, in which the rows indicate the current endpoints, the columns the next ones and in each cell it is presented the value of the distance to go from an endpoint present in the row k to an endpoint present in the column l . When there is no relation between two endpoints, the cost of this edge is equal to infinity. With the cost matrix, the Dijkstra's algorithm can be applied with the help of two vectors which contain the visited, $\mathbf{e}^v$, and the unvisited nodes, $\mathbf{e}^u$. The steps of this algorithm are the following, starting to label all nodes as unvisited ones:

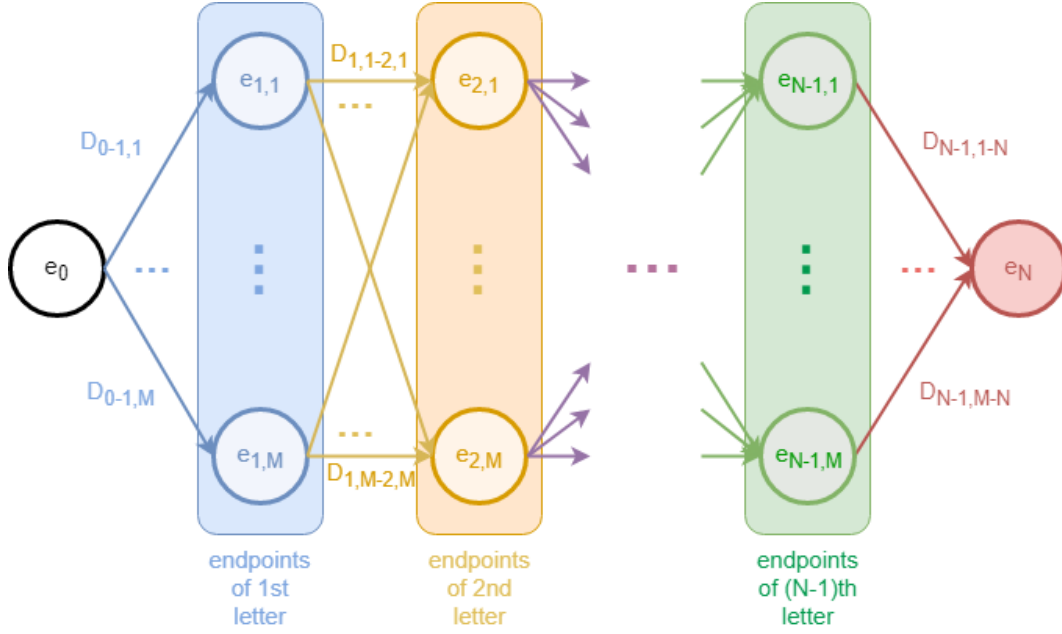


Figure 5.13: Shortest-path algorithm diagram.

- Initialization of vector of distances, $\mathbf{D}$, with size E which is going to contain the distances between the initial node and the rest of the nodes.
- When starting, the initial node, in this case the state e_0 is visited with cost 0. This node is labeled as visited.
- Evaluation of the latest node including in the visited vector, $\mathbf{e}^v$, and labeled as current node, $\hat{e}$.
- Calculation of the tentative distance between each node in the unvisited vector, $\mathbf{e}^u$, which are connected and the current node as follows: $d(e^u_i) = D_{\hat{e}} + d(\hat{e}, e^u_i)$, where $D_{\hat{e}}$ is the distance that node $\hat{e}$ already has in the vector $\mathbf{D}$ and $d(\hat{e}, e^u_i)$ is the distance of going from the current node, $\hat{e}$, to its neighbor unvisited node e^u_i .
- The distance from the initial node to the node e^u_i in the distance vector $\mathbf{D}$ will be: $\min(d(e^u_i), D_{e^u_i})$.
- When all neighbors of the current node, $\hat{e}$, have been analyzed, the current node is labeled as visited, stored in $\mathbf{e}^v$, and removed from the unvisited nodes, $\mathbf{e}^u$.
- The next current node will be that which has the lowest value in vector $\mathbf{D}$.
- Back to step 3 until $\mathbf{D}$ will be completed.

The vector which contains the visited nodes, $\mathbf{e}^v$, at the end of the algorithm must be

contain all nodes of the graph and in each node is marked its precedent one from it can be reached.

In the previous examples, once it has been applied Dijkstra's algorithm in order to find those nodes which minimize the total cost of the path formed by the distances of going from one endpoint to another, the result can be observed in Figure 5.14 which are presented the distance graph and the endpoints of using both center of template and distance map as reference, and in Figure 5.15 where the endpoints for both procedures are located in the tracing of the word.

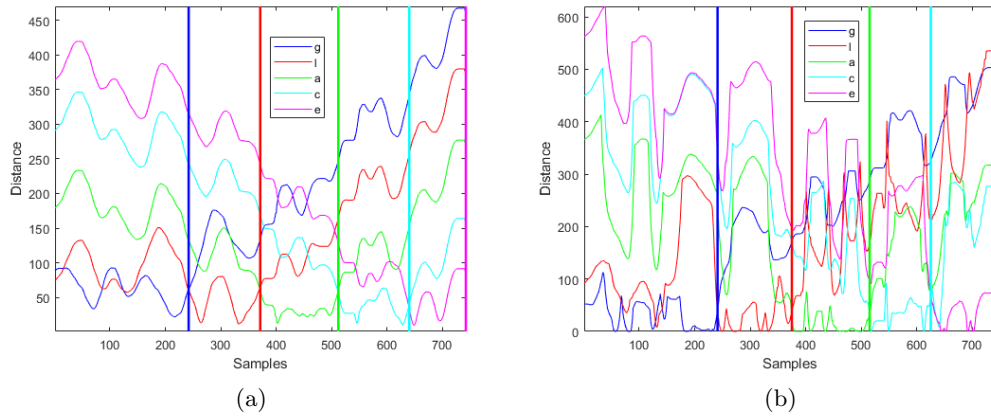


Figure 5.14: Position of the final endpoints of the word "glace" in the distance graph when: a) using CT as reference, b) using the DM as reference.

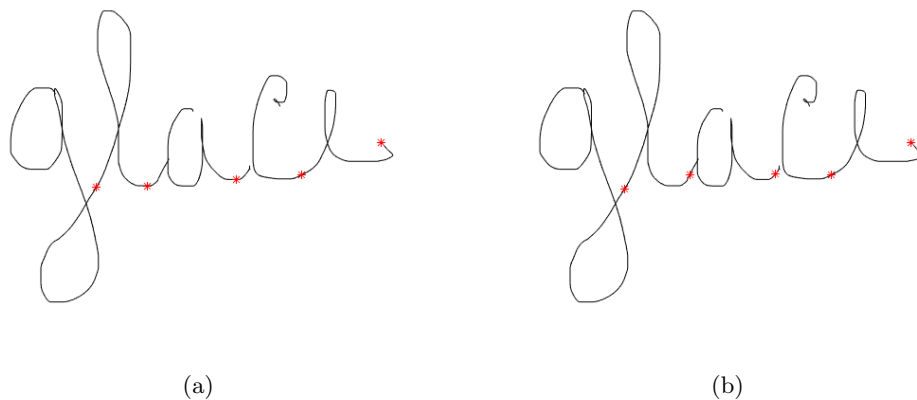


Figure 5.15: Position of the final endpoints of the word "glace" in the tracing of the word when: a) using CT as reference, b) using the DM as reference.

5.4 Special cases

5.4.1 Special characters

In Appendix A the letters used for this project are described. In a cursive writing, where the words are written usually in a concatenated way (see Figure 2.4), there are some characters which necessarily have to be written in two or more steps because they have “subcharacters” as accents, points or other independent strokes, these are “è”, “é”, “â”, “t”, “i” and “j”. Assuming the following aspects in a cursive handwriting:

- First, the main letter structure is written in a consecutive and orderly manner (e.g. the “e” shape of the “é”).
- Secondly, the subcharacters that may be missing are written (e.g. the accent of the “é”).

Taking this into account, the writer has to go back to write these subcharacters which means that the distance between the trajectory of the word and the templates of each letter is no longer linear. This makes the system identify these subcharacters as belonging to the last letter when it is not always the case. In the Figure 5.16 the graphs are not as the ones showed in Figure 5.15 where, looking at the bottom, where the best path is, the minimum distance corresponds to the order of the letters that form the word. In the case of having one of the “special characters” mentioned above, the result is like the one shown in Figure 5.16 corresponding to the word “image” where the point of the “i” is written at the end.

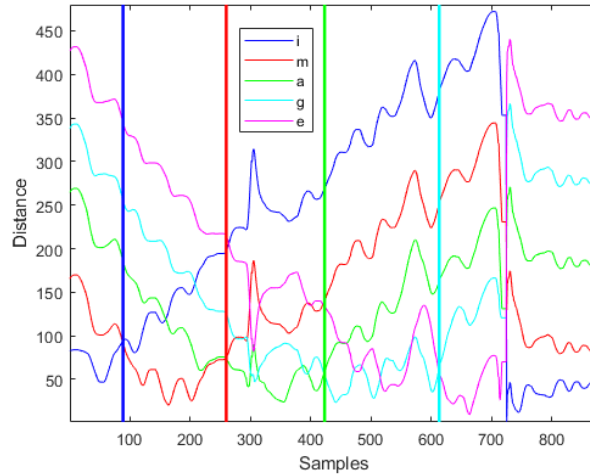


Figure 5.16: Distance graph of “image” without taking into account the point of the “i”.

If this is not taken into account, after performing the segmentation, the subcharacters will appear with the last letter of the word, as it can be observed in Figure 5.17, where it

can be seen how the point of the letter “i” of the word “image” that should be with the letter “i” in first position, appears in the last one with the letter “e”.



Figure 5.17: Result of segmenting the word “image” where the point of the “i” appears with the “e”.

To avoid that, it is necessary to examine the distance graph when one of the letters mentioned appears in the word, considering if there is only one “special character” or more. To develop that, it is defined the following variables:

- C : Number of “special characters” that a word has.
- C' : Number of “special characters” that are not placed in the last position of the word.

Taking these two variables into account, it is defined three different cases, depending the number of C and C' that given word has. To see in which case are the current word, it is defined the schema presented in Figure 5.18.

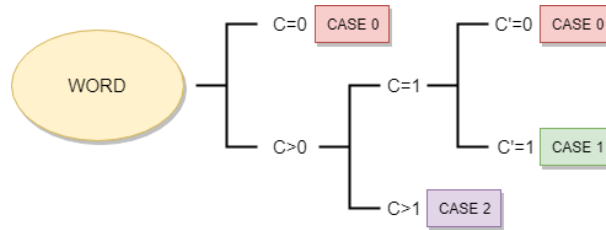


Figure 5.18: Outline of the different cases of endpoint examination that a word can have.

Next, it is going to be explained each of the three cases present in the diagram.

- **Case 0:** In this case, either there is no “special character” or the only “special character” that exists is in the last position of the word. Thus, the way to segment the word into letters is the one shown in the previous section.
- **Case 1:** In this instance, there is only one “special character” in the word which is not placed in the last position of the word. One example of this type of word is “image” and in Figure 5.16 it is shown how at the end of the distance graph the minimum distance corresponds to the letter “i”. In these case, what needs is to

determine the point that the distance of the k -th letter corresponding to the “special character” is lower than the distance of the N -th letter, that is the real endpoint of the last letter:

$$\begin{aligned}
 e_{intersect} &= \{(e_N, e_{N'}) | \xi_k(t) \cap \xi_N(t)\} \\
 \text{subject to} \quad & \xi_k(t = e_N + 1) > \xi_N(t = e_N + 1) \\
 & e_N > e_{N-1}
 \end{aligned} \tag{5.18}$$

In the previous example when it was analyzed the word “image” the results of applying this algorithm are the shown in Figure 5.19.

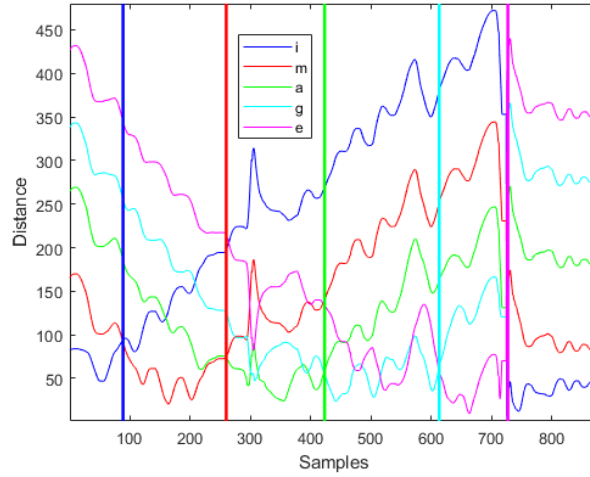


Figure 5.19: Distance graph of “image” taking into account the point of the “i”.

- **Case 2:** In this case, there are several “special characters” in the word. The way to obtain the additional endpoints is similar than the used in the case 1. However, this time it is carry out an iterative procedure in the following way:

1. Creation of the set $\mathbf{C}$ which is the one that contains the $C''=C'$ distance curves associated with the “special characters”.
2. Establishment of a counter $a=0$;
3. Establishment of $\xi_N(t)$ as the first curve of the set, called $\xi_{N+a}(t)$.
4. Intersection between $\xi_{N+a}(t)$ and each one of the C'' remaining curves of $\mathbf{C}$ as follows where c indicates the c -th element of $\mathbf{C}$:

$$\begin{aligned}
 e_{intersect_c} &= \{(e_{N+a}, e_{(N+a)'}) | \xi_{N+a}(t) \cap \xi_c(t)\} \\
 \text{subject to} \quad & \xi_c(t = e_{N+a} + 1) > \xi_{N+a}(t = e_{N+a} + 1) \\
 & e_{N+a} > e_{N+a-1}
 \end{aligned} \tag{5.19}$$

5. Store the C'' x coordinates of the intersection points, $e_{intersect_c}$ in a vector, $\mathbf{e}_{intersect}$.
6. Compute the minimum value of the vector $\mathbf{e}_{intersect}$, which is the new endpoint, e_{N+a} .
7. Increase the counter, $a=a+1$.
8. Remove the c -th distance curve belonging to the new endpoint and establish $\xi_c(t)$ as $\xi_{N+a}(t)$.
9. Return to step 4 until $\mathbf{C}$ is empty.

To see how the algorithm works in this case, it is chosen the word “très” which contains two “special characters”. The resulting distance graph (using CT as reference) is the shown in Figure 5.20.

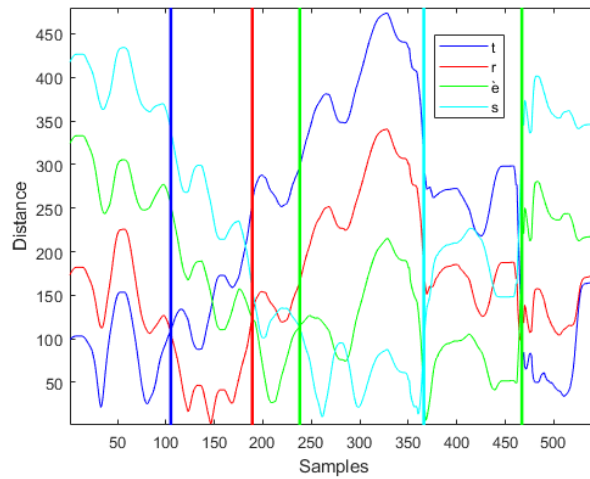


Figure 5.20: Distance graph and endpoint positions after applying the algorithm to the word “très”.

Chapter 6

Results

This chapter presents the results obtained regarding both the letter segmentation proposed in Chapter 5 and the word segmentation performed in Chapter 3. In addition, the definition of the Ground-Truth (GT) of both cases are described.

6.1 Word and line segmentation

6.1.1 Definition of Ground-Truth

The definition of the GT in this case consists simply in labeling each off-stroke of the samples used in the test set.

6.1.2 Data used

To perform this part, given the scarcity of samples and its variability, it has been decided to omit the overlapping strokes and the punctuation marks (periods, commas, colons and apostrophes) with the aim of focusing only on the separation of words in a line determining if the off-strokes are separation points (SP) or not (NSP). The data set has the following characteristics:

$$\text{Total data set: 426 samples} \left\{ \begin{array}{l} \text{Train set} \left\{ \begin{array}{l} \text{SP: 85 samples} \\ \text{NSP: 341 samples} \end{array} \right. \\ \text{Test set} \left\{ \begin{array}{l} \text{SP: 13 samples} \\ \text{NSP: 65 samples} \end{array} \right. \end{array} \right.$$

As observed, the number of NSP in the train set is roughly four times greater than the number of samples of SP, causing the data to be clearly unbalanced and causing this to affect the quality of the results. To solve it, the SPs of the train set have been add three times more. After the addition of the replicate data, the rows of the train data have been shuffled for, as discussed in Chapter 3, have a variety of labels when performing the k-fold

cross-validation.

$$\text{Total data set: 681 samples} \left\{ \begin{array}{l} \text{Train set} \left\{ \begin{array}{l} \text{SP: 340 samples} \\ \text{NSP: 341 samples} \end{array} \right. \\ \text{Test set} \left\{ \begin{array}{l} \text{SP: 13 samples} \\ \text{NSP: 65 samples} \end{array} \right. \end{array} \right.$$

6.1.3 Metrics used

In this case, the metrics used are the following:

- **Confusion matrix** given by the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).
- **Accuracy:** it is the percentage of correct predictions.

$$\text{Accuracy} = \frac{TP + TN}{Total} \quad (6.1)$$

- **Recal:** it is the percentage of positive samples that have been detected correctly.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (6.2)$$

- **Precision:** it is the percentage of detections made that are correct. It is a measure of reliability of the detections.

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6.3)$$

- **F1-score:** harmonic mean between precision and recall.

$$F1 = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \quad (6.4)$$

6.1.4 Word segmentation results

With the data shown in Section 6.1.2, the following results have been obtained with the following configuration of the SVM:

- **Configuration 1:** The best 4 features are used. These features are given by performing the 10-fold cross-validation testing which combination of features gives the best accuracy. It was decided to use 4 features because in the previous tests it gives a very good precision as it is shown in section 3.3.2. The best combination of features is the following:

- **F1**: Length of the off-stroke.
 - **F2**: Angle of the off-stroke with respect to the X axis.
 - **F3**: Distance between the centroids of $B_{s.all}$ and $B_{p.all}$.
 - **F7**: Angle between the vector from the centroids of B_s and B_p and X.
- **Configuration 2**: All the features are used (9 in total) since the combination of them gives a good result at it is shown in section 3.3.2.

For both configurations, as mentioned in Chapter 3 it was selected a RBF kernel. Table 6.1 presents the confusion matrix of both configurations.

		Prediction	
		Positive	Negative
GT	Positive	TP: 10	FN: 3
	Negative	FP: 3	TN: 62

		Prediction	
		Positive	Negative
GT	Positive	TP: 11	FN: 2
	Negative	FP: 4	TN: 61

Table 6.1: Confusion matrix for: left) 4 features, right) 9 features.

In Table 6.2 the results of the metrics described above are presented.

Metric	Configuration 1	Configuration 2
Accuracy	0,9231	0,9231
Recall	0,7692	0,8462
Precision	0,7692	0,7333
F score	0,7692	0,7857

Table 6.2: Results of the SVM model.

As it can be observed, the results of the SVM are really good, since the metrics are really high. To see this, Figure 6.1 shows the ROC curve of both configurations, this curve tells how much the model is capable of distinguishing between classes plotting the True Positive Rate against the False Positive Rate[26]. The AUC is the area under the curve and higher the AUC, better the model is predicting. The values of AUC obtained in the established configurations are the following, which indicate that the model works very well, since the AUC is closer to 1:

- **Configuration 1**: 0,9385.
- **Configuration 2**: 0,9621.

Figure 6.2 presents the results after performing the word segmentation in one line of the test set using the first configuration compared with the GT. As it can be observed, there is a FP in the off-stroke which connects the words “du” and “zoo” causing undersegmentation.

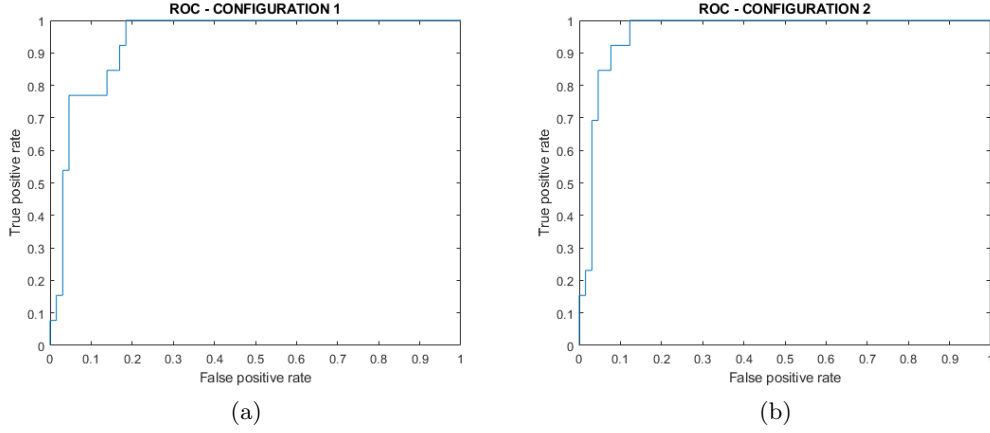


Figure 6.1: ROC of the a) first configuration, b) second configuration.

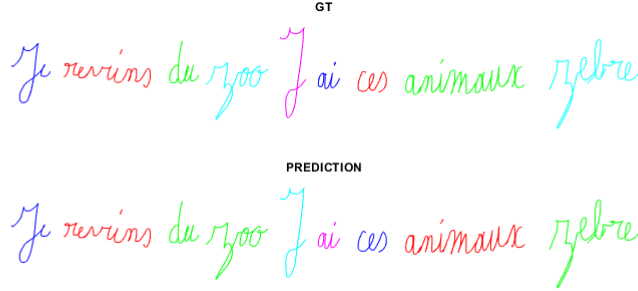


Figure 6.2: Results after performing the word segmentation. Top) GT. Bottom) Prediction.

6.2 Letter segmentation

6.2.1 Definition of Ground-Truth

To measure the performance of the letter segmentation, it is necessary to define a Ground-Truth which indicates in a given word the timestamps corresponding to the endpoints of each letter. Since the typology used in the handwritten data is cursive, the letters of the same word are linked together, which can result in there not being only one correct solution. As it can be seen in Figure 6.3 where the word “glace” appears, there may be several points of the path considered as endpoints of each letter, which are those in red in that figure. This set of endpoints constitute the GT of the letters of the word “glace”. As it can be observed, there is no GT of the last letter since in normal cases, the last sample is considered to be the end of the letter.

Thus, the GT is selected manually as the set of points that can be an endpoint of the letter. To select an appropriate set of endpoints, the shape of the letters of the templates

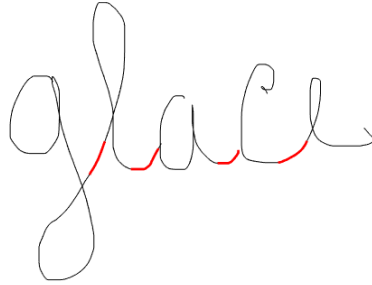


Figure 6.3: Possible set of endpoints of each letter in the word “glace”.

shown in Figure A.1 of Appendix A has been taken into account.

Another point to bear in mind is the detection of what has been called during this thesis “special characters” which are those characters that have to be written in two steps, despite writing with a cursive type. These characters are “é”, “è”, “â”, “i”, “j” and “t”. In this cases the GT it was also defined as the set of points which belongs to a off-stroke (remember that an off-stroke is the set of samples that join two strokes). An example of this GT it is shown in Figure 6.4 where the word “image” appears.

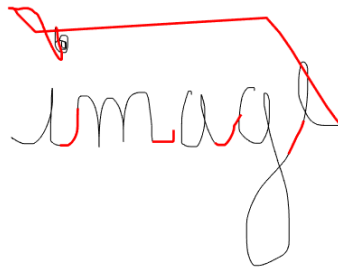


Figure 6.4: Possible set of endpoints of each letter in the word “image”.

6.2.2 Metrics used

With the GT defined, it is necessary to establish several metrics to validate our methodology. It should be noted that as in some cases the endpoint obtained does not fit into the GT but stays closed, it is not considered that the algorithm has failed and that it has segmented poorly, but the error is calculated. To calculate the error, three metrics were defined, two related to the number of samples and the third relative to the Euclidean distance.

Samples error

These metrics indicate the number of samples between the result obtained and the closer point of the GT, Δt . This error can be defined in two different ways, the first one is to present the error as such, that means, Δt , and the other way is calculating the relative error, that is, for each word, it is computed the ratio between the samples error between the k -th endpoint and its GT, Δt_k , and the total number of samples of the word, S . Thus, for a given word with N letters and C' “special characters” which are not placed in the last position (remember that it has been to be defined the separation of the subcharacters, see Figure 6.4), the relative error is computed as follows:

$$error(\%) = \frac{100}{S} * \sum_{k=1}^{(N+C')-1} \Delta t_k \quad (6.5)$$

where

$$\Delta t_k = \min(\Delta t_{k_{start}}, \Delta t_{k_{end}}) = \min(|e_k - GT_{k_{start}}|, |e_k - GT_{k_{end}}|) \quad (6.6)$$

If the endpoint is inside the GT, the error committed is null. In Figure 6.5 it is presented an example of distance between the calculated endpoint and the GT.

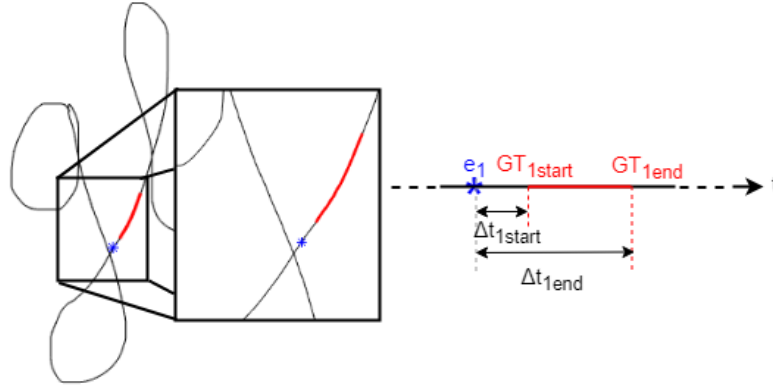


Figure 6.5: Representation of $\Delta t_{l_{start}}$ and $\Delta t_{l_{end}}$ in the first letter of “glace”.

With this, the total error in both cases is computed as the mean between all the values obtained.

Euclidean distance

In this case, it is considered the Euclidean distance between the endpoint and its corresponding GT, when the point is not in the GT. Following the previous scheme, for those

points that are not into the GT, the distance between the endpoint and the GT is defined as follows:

$$d_k = \min(d_{k_{start}}, d_{k_{end}}) = \min(d(f(t = e_k), f(t = GT_{k_{start}})), d(f(t = e_k), f(t = GT_{k_{end}}))) \quad (6.7)$$

where the function $d(P_1, P_2)$ which is the distance between two points is defined as:

$$d(P_1, P_2) = \sqrt{(x_{P_1} - x_{P_2})^2 + (y_{P_1} - y_{P_2})^2} \quad (6.8)$$

As in the previous case, the final distance error is computed as the mean between all the values obtained.

6.2.3 Letter segmentation results

For the results, 20 different words have been chosen with or without special characters, of which half are reference words, that is, they were written by the same person who had written the templates (see Appendix A), and the other ten are words written by children. Most of the words chosen coincide in both cases. It is necessary to clarify that those words that have overlapped letters have not been considered because, as explained in section 6.3.2, the proposed algorithm is not optimized to deal with those cases.

In Figure 6.6 are presented the results relative to the samples error and Figure 6.7 the results relative to the Euclidean distance, both for the case in which the method used to perform the segmentation was the center of the template, *CT*, and for the case in which the distance map, *DM*, was used as reference.

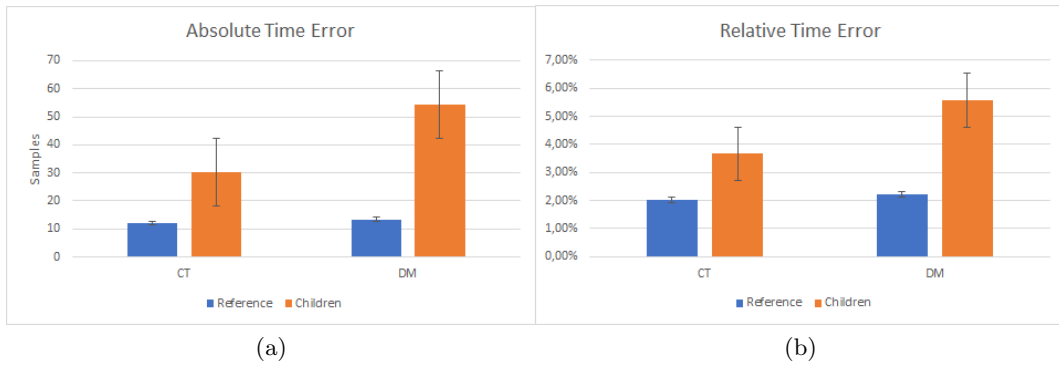


Figure 6.6: Samples error of using *CT* and of using *DM* as reference: a) absolute error, b) relative error.

As it can be observed, the results are considerably better when it is used the reference data since the typology of the templates is the same as that used in the handwritten words.

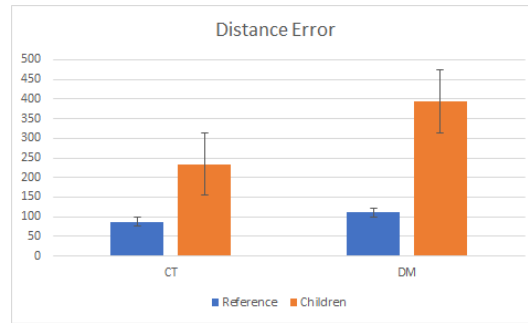


Figure 6.7: Distance error of using *CT* and of using *DM* as reference.

In addition, analyzing some examples of words written by children, it is seen as sometimes between one letter and another there is a pause in time, causing the error to increase, as can be seen in Figure 6.8 where in (a) appears the tracing of the word in time and in (b) it is only shown the strokes written on the tablet. It can be observed how between the “a” and the “v” there is a large trace which makes the error greater.

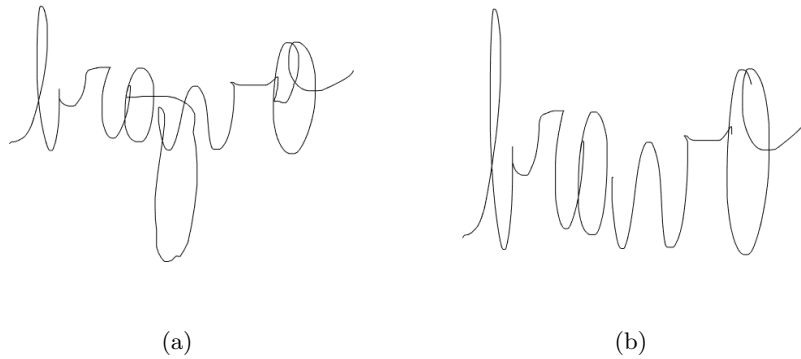


Figure 6.8: Tracing of word “bravo”: a) all the samples recorded by the tablet, b) strokes of the word.

Anyway, looking at the results obtained, it can be seen how the relative error is low (between 2 % and 5 %) which indicates that the proposed procedure works well in most of the defined cases. The other two metrics allow comparing how the two different methodologies defined work under the same conditions, as see, using the center of the template provides a lower error than using the *DM* as reference.

6.3 Weak points

6.3.1 Detection of omitted letters in a word

As mentioned in Chapter 1, one of the effects that a learning difficulty can cause during writing is the bad spelling of the words which can sometimes lead to the omission of letters during the writing of a word. The algorithm presented in Chapter 5 considers always that the words are well-written, taking one template per letter even if the handwritten word has more or less letters than the word to analyze. In this section it is presented a possible procedure to detect when a word has an omitted letter and the way to solve it.

Continuing with the example of this manuscript, two cases are going to be presented, although the procedure has been tested with more examples. The cases are the following:

- **Case 1:** The original and the written word are the same, “glace”.
- **Case 2:** The original word is “grace” while the written one is “glace”. There is an omitted “r” in third position.

As it was explained in Chapter 4, to find the best position of each one of the templates using Lucas-Kanade template warping, it was used a shortest-path algorithm in order to find the positions of the template which minimize the total cost. The cost is formed, as explained in Chapter 4, by:

- NCC cost of placing the k -th template in a pixel of the SW of the k -th letter.
- Distance cost between the reference points of two consecutive templates, the k and the $(k+1)$ -th templates.

Bearing this in mind, it is normal to think that whether there is a misspelling or if there is an omission of a letter in the writing of the word, the total cost will be grater.

In order to find if there is an omitted letter in a given word, it was performed an analysis of the cost given by the solution of the shortest-path algorithm considering both the NCC cost and the cost given by the distance between templates and also considering both costs separately. To put this into an example, it has been chosen the word “glace” and the results when the word is well-written are shown in Table 6.3. In that table, for example, G-(G-L) indicates the NCC cost of the letter “g” and the distance cost between the template of the letter “g” and the template of the letter “l”.

When the original and the written word are different, the cost given by the shortest-path is the shown in Table 6.4. In that case, the original word is “grace” and the written one is “glace”, that means, there is an omitted letter between the “l” and the “a”.

As shown, the results obtained by the cost when using the shortest-path algorithm do not allow to know the position of the omitted letter nor to know if there is an omitted letter.

Original word: GLACE. Written word: GLACE					
Metric	G-(G-L)	L-(L-A)	A-(A-C)	C-(C-E)	E-SINK
NCC + distance	1,5367	2,1748	1,8105	1,7707	1,9707
NCC	0,7613	2,0316	1,3133	1,3225	1,9699
Distance	2,1105	2,2971	2,3022	2,1105	1,9915

Table 6.3: Costs given by the shortest-path algorithm when the original word and the written one are the same.

Original word: GLRACE. Written word: GLACE						
Metric	G-(G-L)	L-(L-R)	R-(R-A)	A-(A-C)	C-(C-E)	E-SINK
NCC+Distance	1,6061	2,4230	1,9500	2,6653	2,7119	2,2854
NCC	0,9233	2,3369	1,5989	2,1461	1,0208	2,2730
Distance	2,2937	2,1166	2,4128	2,3724	2,4485	2,2854

Table 6.4: Costs given by the shortest-path algorithm when the original word and the written one are not the same.

Taking this into account, it is analyzed the way in which the cost is calculated. As explained in Chapter 4, both the value obtained when calculating the NCC between the image and the template and the distance between templates have an associated cost function. But first, to understand the behavior of that function it is important to know the positions of the warped templates given by the Lucas-Kanade algorithm since, as explained, these positions determine the SW in which the correlation and the distance between templates are computed.

In the first example, when the original and the written word are “glace” the best position and the analysis area to compute the NCC cost of the warped templates obtained after LK algorithm is the shown in Figure 6.9 (a), where for each letter, the AA is the bigger area.

In the second case, when there is an omitted “r” in the handwritten word, the position of the AA and the position of the warped template resulting of the LK algorithm is the presented in Figure 6.9 (b), where the position of the missing “r” are marked in yellow.

As it can be observed, in the first case the AA of each letter occupies all the letter, so the NCC will be quite high, especially when the template is positioned in the middle since the letters are in the middle of the AA. On the other hand, in the second case, the AAs do not cover the entire letter, causing the correlation to be lower and the higher correlation values not being those when the template is placed in the middle of the AA. Also, as seen, the omitted letter “r” occupies the position of the written “a” while the “a” is located in the link between “a” and “c”.

To understand the different values of NCC between an image and a template, the curve map that show the NCC when the template is positioned in all the possible places of the

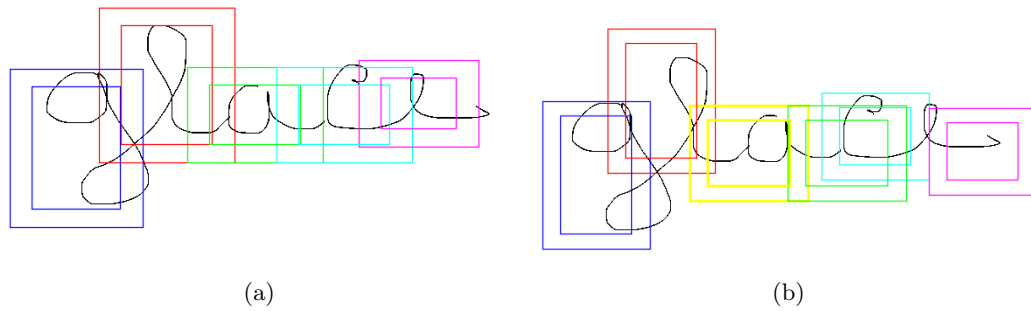


Figure 6.9: AA area and best template position after applying the LK algorithm in: a) “glace” when the original word is “glace”, b) “glace” when the original word is “glrace”.

AA will be presented for both cases, in Figure 6.10 for the first case and in Figure 6.11 for the second. In the curve maps are presented the value of the NCC in each allowed position of the AA, which are the positions of placed the upper-left corner of the template in the SW, as explained in section 4.2 .

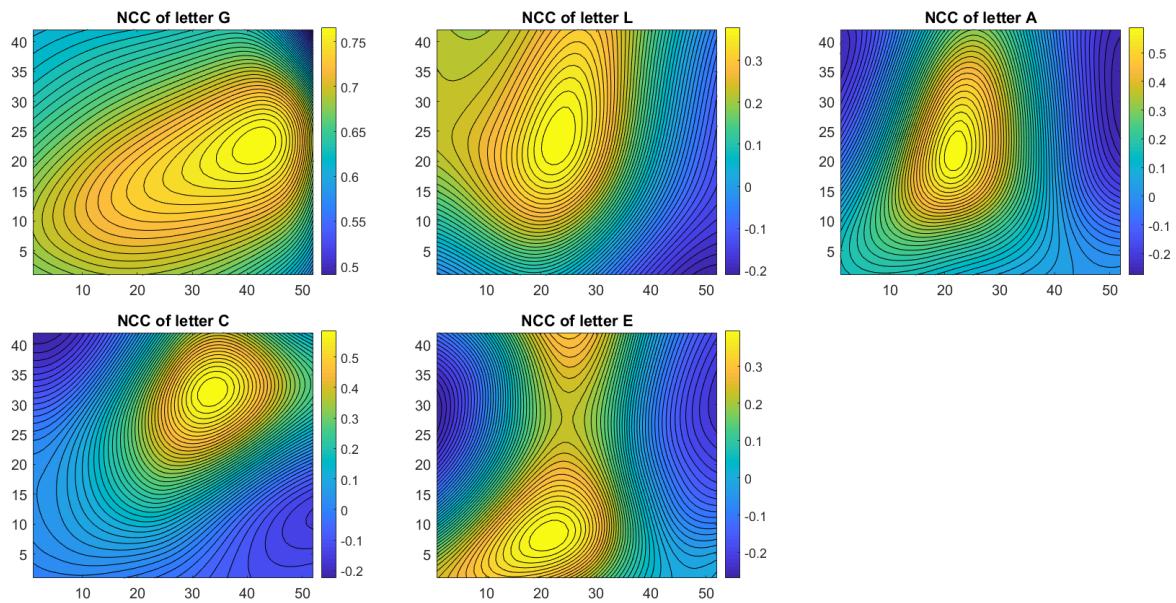


Figure 6.10: Curve map of the NCC when the template is placed in the allowed positions inside the AA when the original and the written word are “glace”.

As seen in the graphs, in the first case, the higher values of NCC are placed in the middle since the AA cover the entire word. However, in the second case, it depends on the letter. Regarding the correlation values, it is seen that when there is a good match between image and template, are greater than 0,3 whereas in the worst cases the correlation is between -0,2 and 0,3. So, these values have to be penalized with the cost function. The current cost function is the one shown in Figure 4.10 (a).

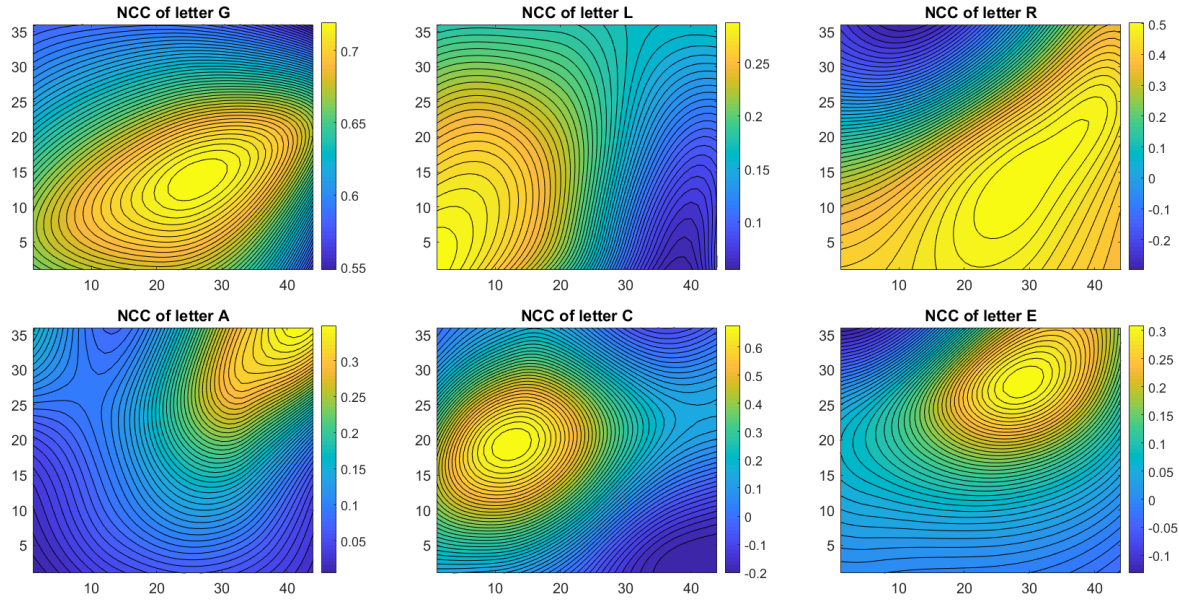


Figure 6.11: Curve map of the NCC when the template is placed in the allowed positions inside the AA when the original word is “glrace” and the written one is “glace”.

By analyzing the total generates to place each template in each of its allowed positions of the AA, it can be observed the probability density functions of the different costs in Figure 6.12 (a) for the case in which the original and the written word are “glace” and in Figure 6.12 (b) for the case in which the original word is “glrace” and the written one is “glace”. In these graphs it has been taken into account both the NCC cost and the cost provided by the distance between templates.

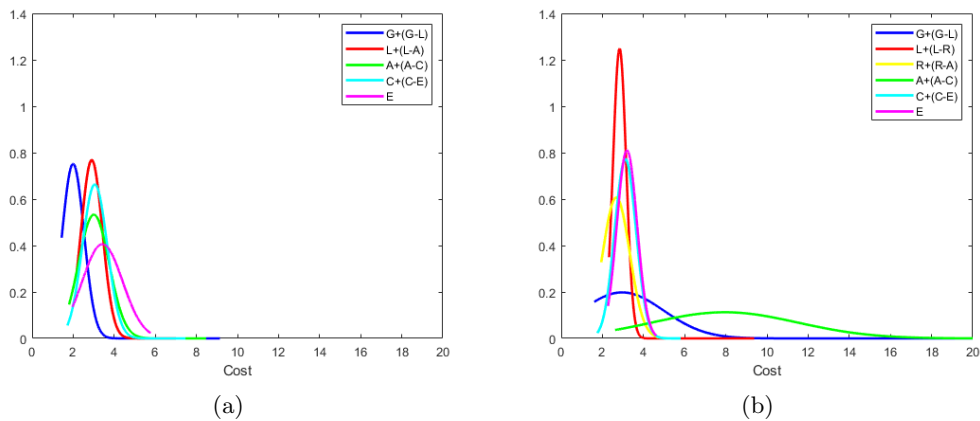


Figure 6.12: Probability density functions in the case of a) the original and the written word is the same (“glace”), b) the original word is “glrace” and the written one is “glace”.

As observed in Figure 6.12, most of the costs are distributed between the values 2 and

4. However, in Figure 6.12 (b), it is shown how the cost corresponding to the NCC cost of the letter “a” and the link between “a” and “c” has higher values. In Figure 6.9 (b) is observed again, it can be seen that this is because after applying the LK algorithm with the graph-based alignment, the final position of the template of “a” is between the written “a” and “c”, so the NCC correlation is very low (see Figure 6.11) and causes higher cost. The problem is that in those cases where the correlation is too low, the associated cost is not always high, as in the previous case where it can be observed that there are samples with very few cost, making the resulted cost of the best-path very low, as seen in Table 6.4.

To avoid that, it was created another cost function for the NCC cost that penalize those values of NCC which indicates a bad correlation, that is, the values below $NCC=0.3$. This cost function is the one shown in Figure 6.13, which is a piecewise function where the values between -0.2 and 0.3 are strongly penalized.

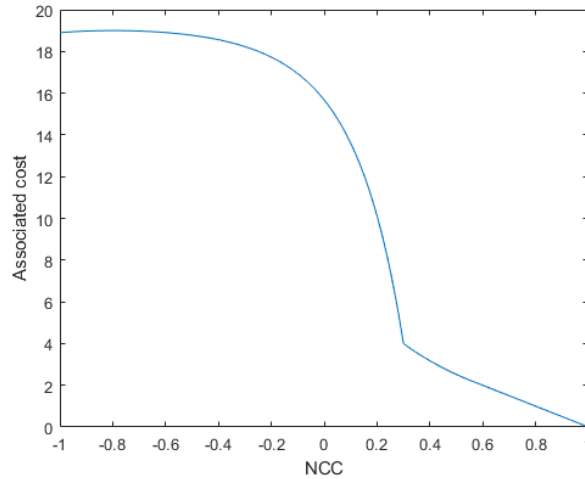


Figure 6.13: New cost function for the NCC.

With this function, the results of the best-path are computed again, obtaining the results shown in Table 6.5 and Table 6.6 corresponding to the both cases described.

Original word: GLACE. Written word: GLACE					
Metric	G-(G-L)	L-(L-A)	A-(A-C)	C-(C-E)	E-SINK
NCC + distance	1,7239	2,8028	2,1417	2,1129	3,1164
NCC	1,1430	3,2504	1,9720	1,9857	3,1149
Distance	2,1105	2,2971	2,3022	2,1105	3,1616

Table 6.5: Costs given by the shortest-path algorithm when the original word and the written one are the same using the new cost function.

As it can be observed in the Table 6.6 when using both the NCC cost and the distance cost the cost associated to the “a” and the link between the “a” and the “c” is higher than

Original word: GLRACE. Written word: GLACE						
Metric	G-(G-L)	L-(L-R)	R-(R-A)	A-(A-C)	C-(C-E)	E-SINK
NCC+Distance	1,8977	3,5410	2,4482	5,1927	3,1282	3,8738
NCC	1,3863	4,5781	2,4103	3,5182	1,5328	3,8406
Distance	2,2937	2,1166	2,4128	2,3724	2,4185	3,8738

Table 6.6: Costs given by the shortest-path algorithm when the original word and the written one are not the same using the new cost function.

the rest. As it can be commented previously, that is why the template is placed in the link between those letters. However, that is not the position of the omitted letter nor the cost is high enough with respect to the rest to determine that there is an omitted letter.

At this point, several questions that could be analyzed in the future in order to find a way to detect missing letters in a word.

- Increase in the AA to calculate the NCC in a wider area that provides more global vision of the surrounding of where the template was located after applying the LK algorithm. This would allow to visualize more exactly where the template would have to be placed to obtain the maximum correlation with the word image.
- Optimize the cost function associated with both the NCC and the distance between templates to greatly favor good matches and penalize when the correlation is low.
- Implement another correlation function instead of NCC. Currently, the method used in section 5.2 has been tested, which consists in calculating in each AA the distance between the curve that defines the word inside the AA, $f(t)$, and the template. However, the results have not been satisfactory since the distance is not a standardized measure and this makes it difficult to compare some results with others to define a good cost function.

6.3.2 Detection of overlapped letters

In same cases, during writing, it is turned back in order to rewrite or correct some parts of the word it was being written. In those cases, the minimum distance provided by the distance graph ceases to correspond to the letters in order (or “special characters”) appearing a new unexpected minimum. To exemplify this, in Figure 6.14 (a) the word “chambre” is presented where it is clearly seen how a part of the letter “b” is rewritten. Figure 6.14 (b) presents its corresponding distance graph using the CT as reference, and it is seen how a part of the letter “b” has been rewritten at the end of the writing. The proposed algorithm of this thesis, knowing *a priori* the number of endpoints that a given word has, is unable to solve these cases. A possible solution would be to always analyze the final part of the distance chart to see if something has been rewritten.

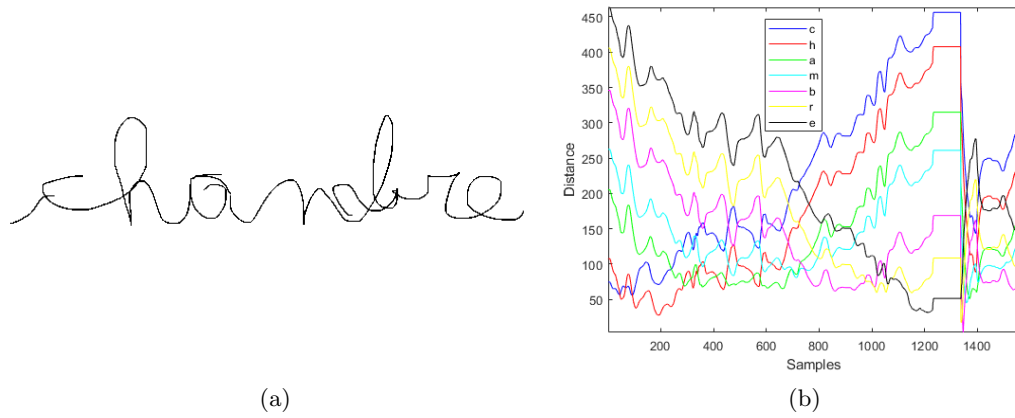


Figure 6.14: Example of overlapped letter where a part of the letter “b” of “chambre” is rewritten: a) image of the word “chambre”, b) its associated distance graph using the *CTs* as reference.

6.3.3 Similarities in the appearance of two letters

Sometimes the segmentation fails, because the position provided by the LK algorithm is not correct since the template is placed on a character similar to itself. An example of this is the sequence of characters “mu” that, in a cursive typology “creates” a fake “u” in between the two letters. This can be seen in the word “mulo” presented in Figure 6.15 where the final position of the letters also appears. As it can be observed, the template corresponding to the letter “u” has been placed in the link between the “m” and the “u” since it has the appearance of a “u”. Also, in this particular case, the first part of the “m” is quite large, seeming that the “m” is formed by three arcs instead of two.

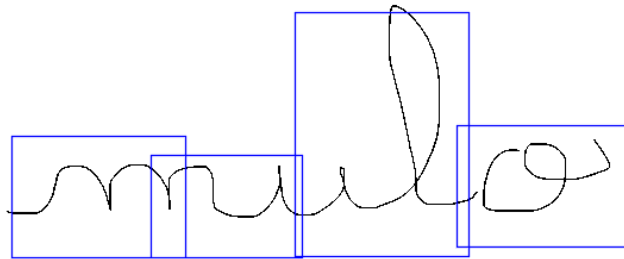


Figure 6.15: Final position of the letters obtained after applying the LK algorithm and the graph-based letter alignment to the word “mulo”.

Chapter 7

Conclusions and further work

7.1 Conclusions

This thesis had two main objectives, on the one hand, given a series of samples corresponding to texts written by children recorded by a tablet as a function of time, be able to segment that text into smaller units, from the higher levels relative to the segmentation into lines of text until the segmentation into words. For the segmentation in lines of text (and in pages if there were any) a method based on the analysis of the data along the time has been proposed. With the position that the data in both its vertical and horizontal coordinates have along the time it can be estimated where a page or a line of text begins, analyzing the shape that the curve acquires over the time. However, for word segmentation, the procedure is not only based on the analysis of data over time. To achieve that, it was used a binary classifier that identifies the union of two strokes, called off-stroke, as a separation point (SP) between words or not. The classifier chosen to perform this task has been an SVM. For its training, it was used nine features extracted from the off-strokes, and it has been validated using the 10-fold cross-validation. The results obtained determine that the resulting model is a good one that allows to perform the word segmentation with good results, as could be seen in Chapter 6.

On the other hand, the other goal of this thesis was that, from the position of the letters in a word obtained after applying a method proposed by [1] based on the combination of the Lucas-Kanade algorithm and the graph-based letters alignment which first warps reference letter templates and then defines an optimal sequence of reference templates and warping parameters, based on a shortest-path algorithm, be able to find the best set of endpoints of each word that allows to segment the word into its letters in an optimal way. For this, a minimization problem has been posed that allows obtaining, from the function of time that defines the word and the position of the letters with its warped templates, the endpoints of each of the letters that minimize the total distance between that function and each of the templates of the letters along the time. To do this, two different strategies have been

implemented, one that takes as reference the center of the template and another that takes as reference the distance map associated to the reference warped template of each of the letters. Apart from that, it has been implemented another method that allows detecting those characters that are necessarily written in two steps. The results obtained determine that this methodology is valid in order to segment words into letter since the relative error obtained over time is not very high (between 2 % and 5 %). However, the algorithm does not work in all the cases, some improvements could still be made to the methodology to avoid the weaknesses pointed out in section 6.3.2, as is the case, for example of the missing letters during the writing.

7.2 Further work

In this section, some lines of further work that have emerged during the development of this thesis are presented:

- Test the effectiveness of the proposed strategy for word segmentation in a larger set of data which also allows performing the segmentation when there are other types of characters such as punctuation marks or apostrophes.
- According to the weak points of the project, presented in section 6.3.2, perform a more exhaustive study of them to be able to achieve a more robust letter segmentation tool which can able to determine if in a word there are missing letters, overlapping letters or even determining if there are misspelled letters. For this, in that section (section 6.3.2), several ideas that could be carried out in the future to improve the performance of the proposed methodology have been introduced.
- Test the performance of the proposed methodology of letter segmentation in a larger set of words that allows observing, for example, how the algorithm works with the same word but written by several different children, or how it acts depending on the number of letters or “special characters” that the words have.

Appendix A

Data used

To develop the project, we have as an input a digital representation of children's handwriting recorded with a pressure-sensitive screen of a digital tablet. This data belongs to different children who wrote some words in French through digital tablets and have been recorded by the Psychology Research team of M. V. Reybroeck and C. Gosse (Faculty of Psychology and Educational Sciences, Université Catholique de Louvain). The data were recorded during a dictation task, the method involved several children in second grade from an ordinary primary school in Belgium's French-speaking region. The device used to record the data was a digital tablet (Wacom Intuos Pro Medium) with a size of 380x251x12 mm together with a digital pen (Wacom Inking Pen). The children were asked to write down the words they heard [5]. The data can be classified in:

- **Texts data:** Data which contain a whole text written by a child.
- **Words data:** Data which contains the information of only one word written by a child.
- **Reference data:** Data which contain the information of the template of each letter of the alphabet.

A.1 Texts data

This data contains several phrases written by three different children. The way to store the data is the following: the user starts to write normally in the upper-left corner of the tablet and continues writing from left to right and from top to bottom, when the page is completely full, the user can write in a new one in the same way. This is important to take into account for the page segmentation since different data appears in the same position of the image.

Feature	Description
Packet Time	Timestamp of each sample in milliseconds.
Packet Serial	Indexing of packets starting at 1 generated by the tablet.
Coordinates	Tridimensional position (X,Y,Z) of the points with respect to the tablet.
Normal Pressure	Pressure of the point on the table. If zero, the pen does not touch the tablet.
Angle	Azimuth and altitude of the pen.

Table A.1: Description of the sentences' data recorded by the.

The phrase written by the children is: *“Je reviens de zoo. J’ai ces animaux : zebre, hippopotame, pithon, bonobo, koala, wapiti, giraffe, orque. Je y avait des beaux oiseaux sur les branches des arbres.”*

The data is structured in the way presented in Table A.1.

A.2 Words data

At this level, there are two types of data. The first one consists in 155 words written by C. Gosse, who has written the reference letters. The second consists in recordings written by 133 different children in primary school who have written the same 48 words (however, younger students wrote only 40 of the 48 words). The typology used in Belgian Primary schools is cursive, which implies joining the letters together by adding links between the letters when writing the links [5]. In Table A.2 it is presented the characteristics of the data recording by the tablet.

A.3 Reference data

The reference data consists in the letters templates used for the letter segmentation. It contains the different letters of the alphabet plus the characters “é”, “è” and “â” written in a identical way than the teachers throughout the French-speaking community of Belgium [5] as it is presented in Figure A.1.

Feature	Description
Index	Indexing of points starting at 0.
Packet Serial	Indexing of packets starting at 1 generated by the tablet.
Slice	Index of the stroke (i.e. sequences of consecutive points that are writing (or not writing)) to which the point belongs.
Writing	Boolean which indicates if the pen is touching the tablet or not.
Group	Index of the word to which the point belongs.
Subject	Id of the user who is writing the words.
Packet Time	Timestamp of each point in milliseconds.
Coordinates	Tridimensional position (X, Y, Z) of the points with respect to the tablet.
Normal Pressure	Pressure of the point on the table. If zero, the pen does not touch the tablet.
Angle	Azimuth and altitude of the pen.

Table A.2: Description of the words data recorded by the tablet.

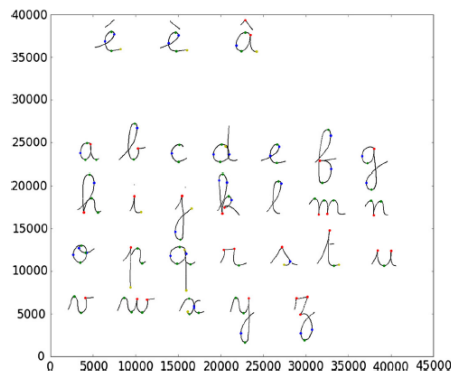


Figure A.1: Reference letters. Image from [5].

Appendix B

Impacts

In this Appendix, it is summarized the impacts that the goal of this project has from different points of view, concretely, from a social, ethic, economic and environmental point of view.

B.1 Introduction

This project is a part of a research carried out by the Psychological Sciences Research Institute and the Institute of Information and Communication Technologies, Electronics and Applied Mathematics, both from the Université Catholique de Louvain. The goal of the study is to gain a better understanding of what constitutes graphic complexity of written material by determining the impact of graphic characteristics on handwriting production. With this study, it is intended to solve the issues related to the interactions between the spelling and graphomotor abilities and the case of children with dyslexia [5].

To develop the project it was necessary to record some data from children, this data consisted of several texts written by children. The participants of the project were twenty children in second grade whose age does not exceed 8 years, with no reported developmental or language delays, from an ordinary primary school in Belgium's French-speaking region. The headmaster and the two teachers gave informed consent, as well as the children's parents. Each child was given a digital tablet and a digital pen to write down the text [5].

B.2 Relevant impacts

According to the context of the project described above, it can be identified several impacts in various areas. First of all, the group of interest of the project is formed by the researches involved in the study, who are, as mentioned before, psychologists and engineers that work together. As said, to develop the project it was used digital tablets in which children from

a school were writing different texts. Thus, one of the groups affected were the children themselves.

Below, some of the most relevant impacts of the project are summarized.

Social impacts

As mentioned, this project is a part of a larger research that studies the interaction between the spelling and graphomotor abilities through the data collected by a tablet when children write on it. The development of a tool that obtains all the features related to the writing of children and the analysis carried out by specialist from the Psychological Sciences Research Institute will allow to find out these unknowns. Moreover, the findings of children's handwritten that the specialists can find with the help of this tool can have practical implications by opening new potential pedagogical guidelines for teaching handwriting at school [5].

Environmental impacts

Regarding the environmental facts, as it was said before, for the experiment it was necessary to bring digital tablets and digital pens to allow the children to write down on the tablets. The rule ISO 14001:2015 [27] describes the life cycle of a product as the consecutive and related stages of a product such as the acquisition of raw materials for its production, the design, the production, the transport, the use and the final treatment of its useful life and final disposal. One way to comply the rule by the institutions which are involved in this project is to re-use the devices after the research, to use them for different purposes to avoid unnecessary waste of plastic and other materials that these devices can contain.

Economic impacts

One of the possible economic impacts of this project can have in the long term, is the cost savings since the development of this tool saves time when carrying out the analysis of texts segmenting them from the highest level (pages, lines) to the lowest level (letters). However, although it saves time by allowing that time to be spent performing other tasks, it requires a maintenance cost because the application may have non-visible defects or because some improvements are desired.

Ethical and professional responsibility

It concerns the children who have participated in the research, as mentioned before, this is one of the affected groups. This impact is the right to privacy and data protection which is collected in the General Data Protection Regulation (EU) 2016/679 [28] and the article

8 “Conditions applicable to child’s consent in relation to information society services” [29] says that the processing of the personal data of a child shall be lawful where the child is at least 16 years old, if the child is below the age of 16 years old, such processing shall be lawful only if and to the extent that consent is given or authorized by the holder of parental responsibility over the child. In the case of this project, the children participating are less of 16 years old so, it is necessary to have the authorization of the parents or legal guardians of each child to avoid economic fines.

Appendix C

Economic budget

In this Appendix it is presented the estimated economic budget of the technical implementation of the project taking into account the material execution expenses, the performers' fees and the taxes involved.

C.1 Direct cost

Material cost

For the execution on this project, it was used an Intel Core i7-3632QM @ 2.2 GHz laptop with 8 GB of RAM and with a SSHD hard disk of 1 TB. Assuming that the computer is new and that it has a useful life of 6 years and that it has been used during the 6 month that the project lasted, the cost is the one summarized in Table C.1. In addition, peripheral equipment such as a mouse and a second additional screen have been used, the cost of which is also shown in that table.

Material cost				
Item	Unit price	Use (years)	Amortization (years)	Cost
Computer	800,00 €	0,5	6	66,67 €
Peripheral equipment	150,00 €	0.5	6	12,00 €
Total				78,67 €

Table C.1: Description of the material cost of the project.

For the development of the project, only a specific software has been used, Matlab R2019a. The university provides free licenses for this software, therefore, the software license cost is considered null.

Labor cost

To compute the labor cost, it is taking into account the hours spent to carry out the project, shown in Table C.2, and the average annual salary of a Telecommunication Engineer published by the COIT [30] which is 52.711,00 €. Assuming that a full-time contract has 8 hours per day and that there are 250 work days in one year [31], there are 2000 work hours per year. Thus, the average Telecommunication Engineer salary per hour is 26,36 €.

Labor cost	
Task	Hours
Study of the state of the art and the project [1]	200
Analysis and design of possible solutions	100
Data preparation	75
Implementation of line segmentation	50
Implementation of word segmentation	100
Implementation of letter segmentation	100
Analysis of special characters	50
Analysis of omitted letters	100
Elaboration of the project report	125
Total	900

Table C.2: Labor cost of the project.

Having worked 900 hours on the project at 26,36 € per hour, the total labor cost of the project is 23.724,00 €.

C.2 Other costs

Overhead expenses

The overhead expenses of the project are the expenses associated to the fact of having an activity in operation. This concerns expenses such as electricity, gas, etc. To calculate it, it has been considered that it is the 10 % of the direct cost of the project, which is formed by the aforementioned costs: material, labor and licensing costs.

With this, the overhead expenses amounts to 2.372,40 €.

Industrial profit

It is computed as the 6 % of the sum of the direct and the indirect (overhead expenses) costs of the project. Thus, the industrial profit is 1.624,38 €.

Consumable material

The cost of consumables is estimated at 50,00 € in terms of photocopies, paper, ink, etc.

C.3 Total cost

The total cost is shown in Table C.3 where it is summarized all the costs described above and where it is applied the 21 % of taxes to the subtotal.

Total cost	
Concept	Cost
Material cost	78,67 €
Labor cost	23.724,00 €
Budget of material execution	23.802,67 €
Overhead expenses (15 %)	3.570,40 €
Execution budget	27.373,07 €
Industrial profit (6 %)	1.624,38 €
Consumable material	50,00 €
Subtotal	29.065,45 €
Taxes (21 %)	6.103,74 €
Total	35.169,19 €

Table C.3: Summary of the economic budget of the project.

Bibliography

- [1] Clara Sansalvadó. *Graph-based segmentation of letters in handwriting words with Lucas Kanade template warping*. Université Catholique de Louvain, 2018.
- [2] Pratishruti Saxena. A review: English online handwriting recognition. *Special Conference Issue: National Conference on Cloud Computing and Big Data*, pages 163–169, 2015.
- [3] Frank de Zeeuw. Slant correction using histograms. *Bachelor's Thesis in Artificial Intelligence*, 2006.
- [4] Homayoon S.M. Beigi. An overview of handwriting recognition. 1997.
- [5] Claire Gosse, Simon Carbonelle, Christophe de Vleeschouwer, and Marie Van Reybroeck. Specifying the graphic characteristics of words that influence children's handwriting. 31:1181–1207, 2018.
- [6] Carmen Díaz Oyarce and María Francisca Price Herrera. ¿Cómo los niños perciben el proceso de la escritura en la etapa inicial? *Estudios Pedagógicos XXXVIII*, 1:215–233, 2012.
- [7] Sonia Kandel and Cyril Perret. How does the interaction between spelling and motor processes build up during writing acquisition? *Cognition*, 136:325–336, 2015.
- [8] Laura H Dinehart. Handwriting in early childhood education: Current research and futures implications. *Journal of Early Childhood Literacy*, 0:1–22, 2014.
- [9] Gombert J and Fayol M. The influence of spelling ability on vocabulary choices when writing for children with dyslexia. *Learning and Instruction*, 2:23–41, 1992.
- [10] Emma Sumner, Vincent Connelly, and Anna L. Barnetti. The influence of spelling ability on vocabulary choices when writing for children with dyslexia. *Journal of Learning Disabilities*, pages 1–2, 2014.
- [11] Emma Sumner, Vincent Connelly, and Anna L. Barnett. Children with dyslexia are slow writers because they pause more often and not because they are slow at handwriting execution. *Reading and Writing*, 26:991–1008, 2013.

- [12] Bob Cunningham. How various learning and attention issues can cause trouble with writing. <https://www.understood.org/en/learning-attention-issues/child-learning-disabilities/writing-issues/how-various-learning-and-attention-issues-can-cause-trouble-with-writing>. Access 26-03-2019.
- [13] Subhais Mandal, S. R. Mahadeva Prasanna, and Suresh Sundaram. Gmm posterior features for improving handwriting recognition. *Expert systems with applications*, 97:421–433, 2018.
- [14] Réjean Plamondon and Sargur N. Srihari. On-line and off-line handwriting recognition: a comprehensive survey. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.
- [15] Anisha Priya, Surbhi Mishra, Saloni Raj, Sudarshan Mandal, and Sujoy Datta. Online and offline character recognition: A survey. *International Conference on Communication and Signal Processing*, 2016.
- [16] Jorge Sueiras, Victoria Ruiz, Ángel Sánchez, and José F. Velez. Offline continuous handwriting recognition using sequence to sequence neural networks. *Neurocomputing*, 280:119–128, 2018.
- [17] Cuong Tuan Nguyen, Bilan Zhu, and Masaki Nakagawa. A semi-incremental recognition method for on-line handwritten english text. *International Journal of Computer Vision*, 2014.
- [18] Cuong Tuan Nguyen and Masaki Nakagawa. An improved segmentation of online english handwritten text using recurrent neural networks. *3rd IAPR Asian Conference on Pattern Recognition*, 2015.
- [19] Amit Choudhary, Rahul Rishi, and Savita Ahlawat. A new character segmentation approach for off-line cursive handwritten words. 17:88–95, 2013.
- [20] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin. A survey of ocr applications. *International Journal of Machine Learning and Computing*, 2, 2012.
- [21] N. Cristianini and J. Shawe-Taylor. *Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [22] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of database systems*, 2009.
- [23] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56:221–255, 2004.

- [24] J.P Lewis. Fast normalized cross-correlation. 1995.
- [25] Álvaro H. Salas S. *Acerca del algoritmo de Dijkstra*. Universidad de Caldas and Universidad Nacional de Colombia, 2008.
- [26] Sarang Narkhede. Understanding AUC - ROC curve. <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>. Access 04-06-2019.
- [27] Iso 14001:2015. 2015.
- [28] European Parliament and Council of the European Union. General data protection regulation (eu) 2016/679. 4 May 2016.
- [29] European Parliament and Council of the European Union. Article 8: Conditions applicable to child's consent in relation to information society services. <http://www.privacy-regulation.eu/en/8.htm>, 4 May 2016. Access 24-05-2019.
- [30] Colegio Oficial de Ingenieros de Telecomunicación. Pleno empleo para los ingenieros de telecomunicación. <https://www.coit.es/noticias/pleno-empleo-para-los-ingenieros-de-telecomunicacion>. Access 06-06-2019.
- [31] Wroking days. <http://www.dias-laborables.es/>. Access 06-06-2019.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl