Chapter 1

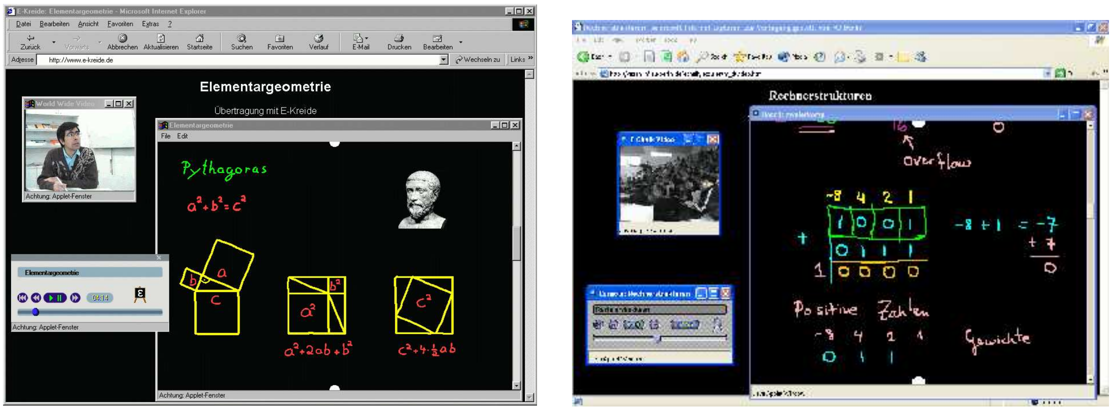
Introduction

1.1 Motivation

Today, computer programs are capable of generating the most complex mathematical texts containing complicated two-dimensional mathematical notation, which can be perfectly understood by humans. However, the inverse is still a complicated process. The mathematical knowledge represented by formulas and mathematical notation, generated by humans for humans, is not easy to transform automatically into an internal computer-processable representation.

As a partial solution to this problem, optical character recognition is a very important technology, because most of the mathematical knowledge is transmitted in printed and handwritten form. There are some partial advances in the field of optical character recognition and its application to recognize mathematical notation. Researchers have come up with different solutions to recognize mathematical formulas in printed scientific texts. For example, one has to scan some book and select the desired text to be processed. Formulas and mathematical expressions will be extracted from the pages and recognized. In this way, recognition of mathematical notation became an important pattern recognition problem.

Other very different scenarios have to be considered when mathematical notation are presented by other means, e.g. with the stylus of an electronic tablet, a Tablet PC, or a personal digital assistant (PDA). For such devices, there are programs capable to process and recognize a determined kind of handwritten input, but not handwritten mathematics. This bring us to ask the following question: When do we have to consider the real-time recognition of handwritten mathematics? The answer to this first question will be described in the next section.



(a)

(b)

Figure 1.1: *The E-Chalk system.*

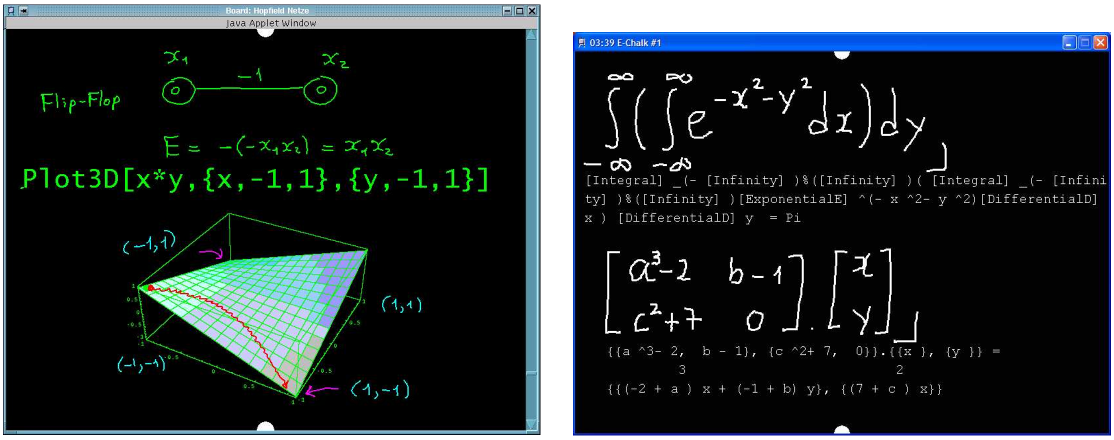
1.1.1 The Electronic Chalkboard

The *Electronic Chalkboard* (*E-Chalk*) is a software system for distance teaching developed at the Freie Universität Berlin [44]. The aim of the system is to improve the live classroom experience by transforming a contact-sensitive screen into a smart teaching tool, see Fig. 1.1(a). The lecturer uses the stylus exactly as he uses a classic chalkboard, but it has the advantage of changing automatically the color of the writing, its thickness, etc.

The software does not simply enhance the readability the lecturer's writing. Due to its Java-based technology it is also possible to show to the students, apart from the lecturer's writing, applets from anywhere on the Web, as well as to use other Web services. The E-Chalk technology also allows the live transmission of lectures. The lecture may be provided live as a stream, which is made available to the viewers on a main server. The viewer needs only to have a Java-enabled web browser installed on his computer and to click on the link to the stream to replay the lectures. See Fig. 1.1(b).

Because the system was first introduced at the Faculty of Mathematics and Computer Science of the Freie Universität Berlin, it was necessary to enable interaction capabilities with computer algebra systems, like Mathematica or Maple. For example, if it is necessary for the lecture to plot a function or to find the solution to some calculus problem, the lecturer can provide algebraic expressions via the keyboard, as shown in Fig. 1.2(a). In this way, a great number of lectures and conferences held at the faculty have been stored, ready to be replayed by interested viewers.

However, this way the system handles mathematics does not follow the philosophy


Board: Hopfield Netze
Java Applet Window
x_1
x_2
-1
Flip-Flop
$E = -(-x_1 x_2) = x_1 x_2$
(-1,1)
(1,1)
(-1,-1)
(1,-1)
03:39 E-Chalk #1
$\int_{-\infty}^{\infty}\left(\int_{-\infty}^{\infty} e^{-x^2-y^2} dx\right) dy$
$\begin{bmatrix} a^3-2 & b-1 \\ c^2+7 & 0 \end{bmatrix} . \begin{bmatrix} x \\ y \end{bmatrix}$

```
Plot3D[x*y,{x,-1,1},{y,-1,1}]
```

```
[Integral] _(- [Infinity] )%([Infinity] )( [Integral] _(- [Infini
ty] )%([Infinity] )[ExponentialE] ^(- x ^2- y ^2)[DifferentialD]
x ) [DifferentialD] y  = Pi
```

```
{{a ^3- 2,  b - 1}, {c ^2+ 7,  0}}.{{x }, {y }} =
         3                          2
{{(-2 + a ) x + (-1 + b) y}, {(7 + c ) x}}
```

(a) (b)

Figure 1.2: *Using mathematics in the E-Chalk.*

which inspired the design of the E-Chalk system: the input should be given only by using the stylus, reducing the interaction through other devices to a minimum. The system should be capable to process handwritten input as shown in Fig. 1.2(b). The interest to provide the E-Chalk system with a recognition tool for handwritten mathematical expressions motivated the work described in this thesis.

1.2 Characteristics of Handwritten Data

Optical character recognition (*OCR*) involves computer systems capable of transforming typeset or handwritten text into computer-processable text. OCR is one of the most studied subjects in the area of pattern recognition and computer vision.

When considering the recognition of text, authors speak about typeset and handwritten OCR. This division considers how the text was *originally* generated, see Fig. 1.3. OCR systems can be divided further by considering the way the *raw data* was obtained by the computer. The raw data is a *digitalization* of the original text. It is done by means of some input device, like a scanner or a contact-sensitive screen. In the following section we discuss the two different kinds of data used by OCR systems.

1.2.1 Off-Line and On-Line Data

Off-line data is given to the computer as an image which is commonly obtained by scanning documents. This is the most common data digitalization process for typeset text. For that reason, most of the difficulties encountered when recognizing typeset

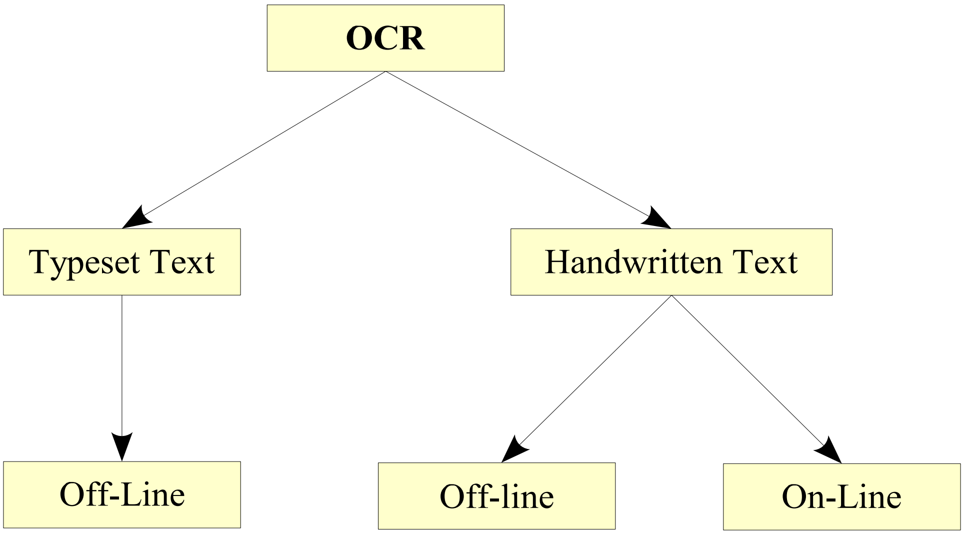


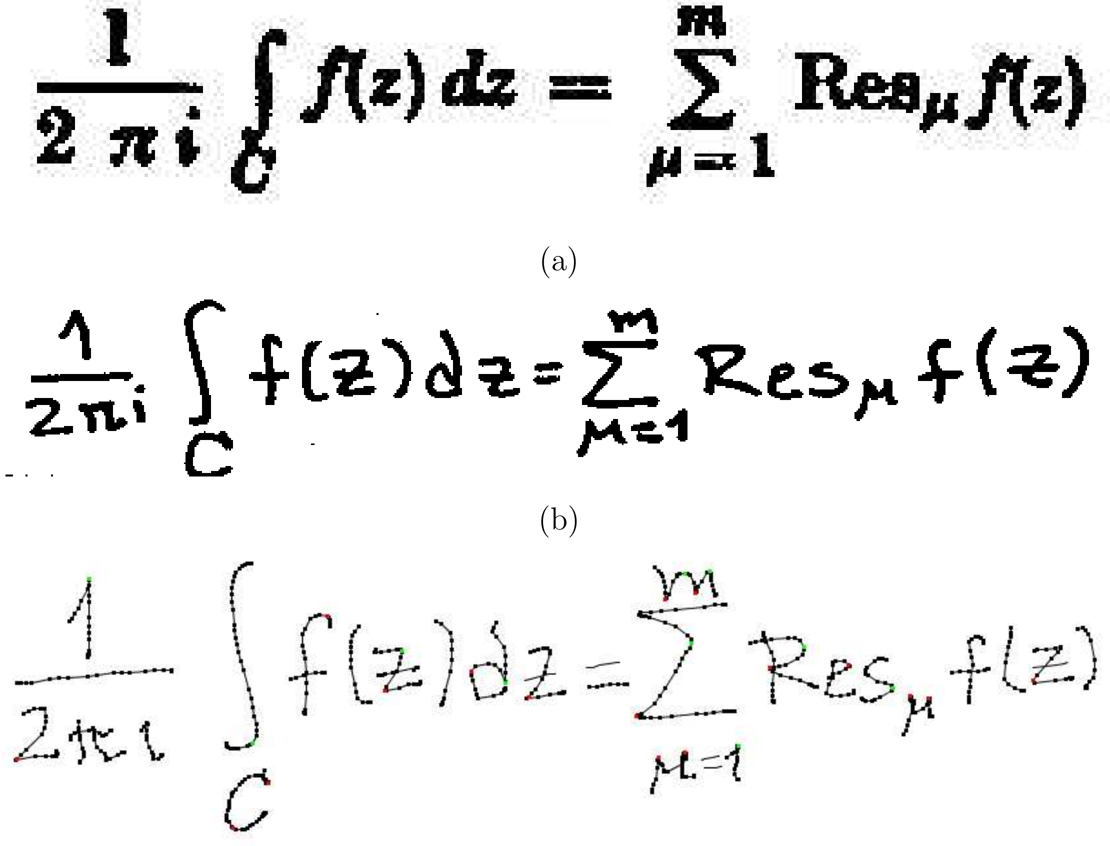
Figure 1.3: *Optical character recognition.*

text remain on the low quality of the scanning and the great variety of font types used in documents. Off-line data is considered as a *static* representation of text, because there is no time stamp stored in it which can describe the order of printing or handwriting. See Fig. 1.4(a)-(b). Generally, the original data was written or typeset some time ago, before the recognition process takes place. The relevance of off-line OCR remains in the fact that it allows the conversion of any typeset document generated before the computer era into a format compatible for computer processing.

In the case of handwriting, the data can also be *on-line*, see Fig. 1.4(c). The raw data considered in on-line recognition are *points* and *strokes*. Points have time-stamp information obtained by the stylus of a digital tablet. They are normally stored at regular time intervals. Strokes are sequences of points generated between *pen-down* and *pen-up* events of the pen device. On-line data is also known as *digital ink*, which is a *dynamic* representation of handwriting. Contrary to the off-line case, the recognition process of on-line data takes place during the writing or immediately after it is finished.

In the E-Chalk system, the handwriting of the lecturer is represented as on-line data. When the viewer replays the lecture, it is shown exactly as the lecturer wrote it, like a movie using the stored time-stamp information. The E-Chalk system can also convert the lecture into a PDF file for easy printing. During the conversion, the time-stamp information is lost. In this way, the file is an off-line version of the lecture.

One of the advantages when dealing with on-line data is the time-stamp information contained in it. For example, time-stamp information facilitates the handling of



(a)

(b)

(c)

Figure 1.4: *Different data representation of a mathematical expression: (a) scanned typeset expression, (b) off-line handwritten expression, and (c) on-line handwritten expression.*

symbol overlapping. When off-line symbols overlap, we can only use the "spatial" distance to separate them. When dealing with on-line data, the temporal information makes any symbol potentially "separable" by using an extra dimension added by the time-stamp. On-line data also offers other useful information, like writing pressure, speed and acceleration. For on-line data, the stroke's thickness does not play any important role. On the contrary, skeletonization (thinning) of off-line data is necessary if we want to obtain acceptable recognition rates.

However, the dynamic representation of writing also has some disadvantages. While off-line symbols, words, and formulas are invariant to the order they were written, that is not the case for on-line data. Most on-line symbols consist of a sequence of strokes which can be written in a different order and direction and will still represent the same symbol. This is one of the most important problems during the

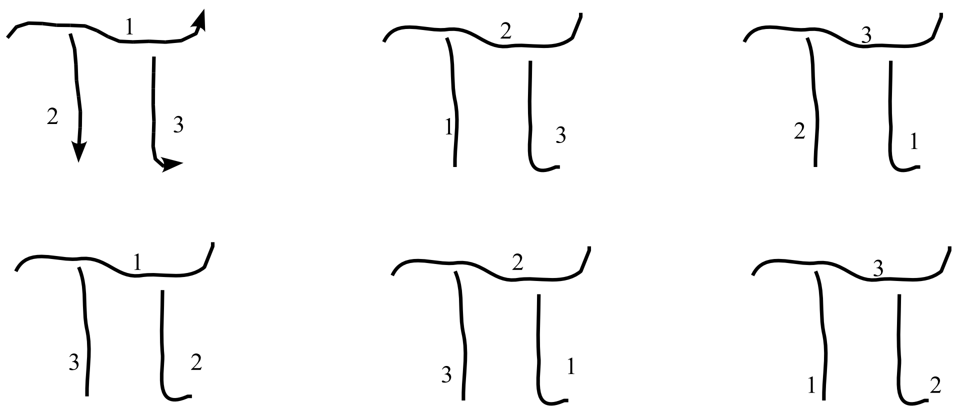


Figure 1.5: *Different ways to draw the symbol 'π' by changing only the strokes' order. All the strokes were written by following the directions indicated by the arrows.*

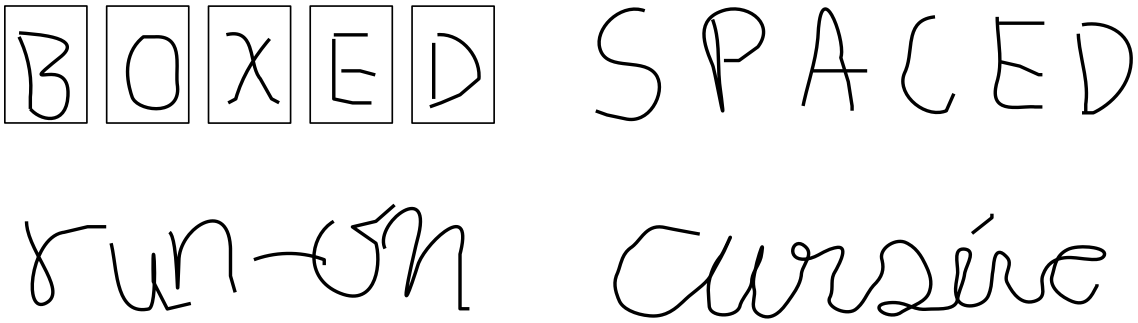


Figure 1.6: *Different styles of handwriting.*

recognition process, as we can see in Fig 1.5. The symbol 'π' can be written in six different ways, only taking into account the different order the strokes can be written in. It gets even harder if we also vary the strokes' direction.

1.2.2 Styles of Handwritten Data

Handwritten data is also characterized by the *writing style* of users [87]. The first style to be considered is the block or isolated style. In this style, symbols, letters, and words are clearly separated by boxes, used as guides, or by leaving a distinguishable space between them. These characteristics correspond to the boxed and spaced styles respectively. In contrast, free style allows more freedom on the writing. Symbols may overlap, share strokes, or even mix these styles. This characterizes run-on, cursive, and mixed styles respectively. See Fig 1.6. The styles which are normally used when writing mathematical expressions are run-on and spaced ones.

Actually, the difficulties encountered in handwriting recognition lie in the freedom the user takes when he writes. The more freedom the user takes, the more difficult the recognition of the data will be.

1.3 Characteristics of Mathematical Notation

Before we proceed to enumerate the necessary steps for the recognition of mathematical formulas, we have first to consider the characteristics of this specialized notation which generate difficulties during the recognition, as suggested by Blostein [9], and Chan and Yeung [14]:

Definition of Mathematical Notation

Mathematical notation is a specialized *two-dimensional notation* which helps to communicate mathematics and to visualize concepts and ideas. Although mathematical notation is a language used in many areas of science, no formal definition in terms of syntax and semantics as a two-dimensional language exists. Actually, this notation is not completely standardized and many *dialects* are used by scientists. Some authors try to describe mathematical notation for solving problems of typesetting [38] and for automatic processing of mathematical notation [45].

Scope of Recognition Systems

We have to consider which kind of mathematical expressions we will recognize. Researchers normally restrict themselves in the recognition of a subset of mathematical notation, for example notation for a particular area of mathematics, notation needed only for high-school mathematics, etc. In addition, such restrictions also depend on the purpose of the recognition system: recognition of typeset scientific text, an input model for computer algebra systems, etc.

Grouping of Basic Symbols

When we read an expression, we group single symbols to construct more complex objects, which plays an important role in the interpretation of formulas. For example, the digits ‘3’, ‘8’, and ‘1’ have their own meaning as such, but if they are of the same size and lie in the same line, they can represent the integer value ‘381’. By varying size and location they can also represent ‘3^{81}’ or ‘$3^{8}1$’. Similarly, we can group some letters to represent function names, like ‘log’ or ‘sin’.

Explicit and Implicit Operators

Variation of size and location can also represent grouping criteria through mathematical relations between symbols. In mathematical notation, we find *explicit* and *implicit* relations between symbols. Subscripts, superscripts and tabular structures are described only by the location of operands; in contrast, addition, subtraction, and division use explicit operator symbols. For example, in the expression '$a + b$' the explicit relation 'addition' between 'a' and 'b' is given by the symbol '+'. An example of an implicit relation would be in the expression 'x^2'. In this case the expression represents the relation 'pow' between 'x' and '2', a variation in the location results in a very different relation, for example 'x_2', representing the relation 'subscript'.

Ambiguity in the Role of Symbols

We illustrate this characteristic by giving some examples. The horizontal bar can represent the minus sign and also the fraction bar. A dot can represent multiplication or the decimal point. The Greek letter sigma 'Σ' can also represent the operator sum. When grouping the symbols 'dy', they can represent the product of 'd' and 'y' as in '$cx + dy$' or the differential operator when used in '$\int \cos(y)dy$'.

Irregular Writing

Irregular handwriting aggravates ambiguities described before and makes it harder to group symbols and to distinguish relations among them. It results in layout problems affecting the recognition of the whole expression. A cause of this is due to unexperienced users, because they normally take excessive freedom with the location and alignment of handwritten symbols. Other kinds of irregular writing arise during the correction, deletion, and insertion of symbols. For example, suppose an expression was entered by the user and he decided to write some super-indices or sub-indices, but there was not enough space available for this correction. Something like this could generate a very complex expression, which could not be recognized even by humans.

1.4 Steps for Recognition of Mathematical Notation

When considering the main characteristics of on-line handwriting and mathematical notation, authors have divided the recognition of mathematical expressions into the

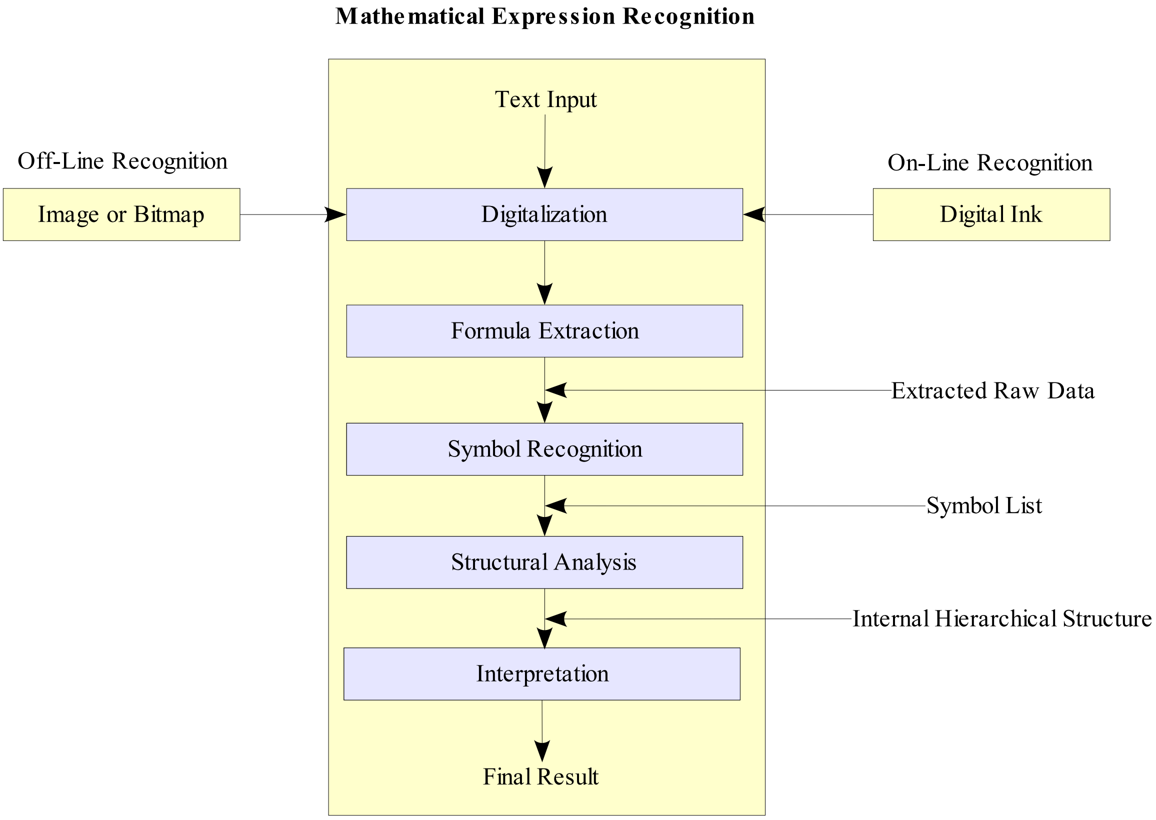


Figure 1.7: *Diagrammatic representation of the stages to recognize mathematical notation.*

stages shown in Figure 1.7.

The recognition of mathematical notation begins by considering how the data is presented. The *digitalization* step transforms the given expression into a static or dynamic representation.

The *formula extraction* step is normally needed when dealing with typeset off-line data. In this step, mathematical expressions are extracted and isolated from text lines [55, 42]. When dealing with on-line data, it is normally supposed that the data only contain mathematical expressions.

Once the extracted raw data is obtained, the next step is *symbol recognition*. In this step, the label of symbols is established by means of a classifier.

During the *structural analysis* step, a list of the recognized symbols is grouped to represent the expression as an internal hierarchical structure suitable for interpretation and processing by a computer program.

The last step is *formula interpretation*. In this step, the internal hierarchical structure is processed and interpreted to obtain a final result. This result can be a character string which represents the expression in LaTeX, another structure used by a computer algebra system, or an image representing the plot of a function given by the expression, among others.

1.5 Objectives and Structure of this Thesis

The objective of this thesis is to address the problem of the recognition of mathematical notation on systems where data is entered via pen-based devices, like graphic tablets, contact-sensitive whiteboards, or Tablet PCs. In particular, we will develop a recognizer of on-line handwritten mathematical expressions for the E-Chalk system. For this purpose, we concentrate on the application of different classification methods for the recognition of the most frequently used symbols in mathematical notation. We also concentrate on the development of a robust method for the structural analysis of mathematical notation.

In **Chapter 2** we give an overview of some methods used for the recognition of mathematical expressions. We review the current and relevant methods for the recognition of *on-line* handwritten mathematical expressions.

The description of the preprocessing methods for on-line data is given in **Chapter 3**. Preprocessing is a preliminary step necessary to eliminate noise, to reduce the amount of information, and to normalize handwriting. The application of preprocessing can dramatically improve the classification results.

In **Chapter 4** we compare different classification methods for on-line handwritten isolated characters. We describe how we considered local and global features to improve the classification results.

In **Chapter 5** we see how structural layout analysis is used to translate the classification results into a tree of relationships among the symbols. We explain how such an analysis is based on a minimum spanning tree construction and symbol dominance.

We developed an editor for on-line handwritten mathematical expressions, the description of which is given in **Chapter 6**. The program generates output suitable for use in word processors, symbolic computation, or certain programming language.