

Automatic extraction of printed mathematical formulas

using fuzzy logic and propagation of context

A. KACEM
ENSI-RIADI-Tunisia
Afef_kacem@yahoo.fr

A. BELAÏD
LORIA-CNRS-France
Abelaid@loria.fr

M. Ben AHMED
RIADI-Tunisia
<mailto:abelaid@loria.fr>
Mohamed.benahmed@serst.rnrt.tn

Abstract

This paper describes a new method to segment printed mathematical documents precisely and extract formulas automatically from their images. Unlike prior methods, it is more directed towards the segmentation rather than the recognition, isolating mathematical formulas outside and inside text-lines. Our ultimate goal is to delimit parts of text that could disturb OCR application, not yet trained for formula recognition and restructuring. The method is based on a global and a local segmentation. The global segmentation separates isolated formulas from the text lines using a primary labeling. The local segmentation propagates the context around the meted mathematical operators to discard embedded formulas from plain text. The primary labeling identifies some mathematical symbols by models created at a learning step using fuzzy logic. The secondary labeling reinforces the results of the primary labeling and locates the subscripts and the superscripts inside the text. Some heuristics has been defined that guides this automatic process. In this paper, the different modules making up the automated segmentation of mathematical document system are presented with examples of results. Experiments done on some commonly seen mathematical documents, show that our proposed method can achieve quite satisfactory rate making mathematical formula extraction more feasible for real-world applications. The average rate of primary labeling

of mathematical operators is about 95.3% and their secondary labeling can improve the rate about 4%. The formula extraction rate, evaluated with 300 formulas and 100 mathematical documents having variable complexity, is close to 93%

Keywords : Mathematic formula extraction, document segmentation, symbol labeling, fuzzy logic, context propagation

1. Introduction

With the ultimate objective of a high-level understanding of mathematical document content, the need for advanced formulas extraction and recognition technologies is on the rise, so as to take full advantage of the semantics conveyed by the formulas as an important and the crucial part of mathematical document. This paper is devoted to mathematical formula extraction.

Formulas are involved in mathematical documents, either as isolated formulas, or embedded directly into a text line. They have a number of features, which distinguish them from conventional text. These include structure in two dimensions (summations, products, integrals, roots, fractions, etc.), frequent font changes, symbols with variable shape and size according to the context (brackets, fraction bars, subscripts, superscripts, etc.), and substantially different notational conventions from source to source. When compounded with more generic problems such as noise and merged or broken characters, printed mathematical expressions offers a challenging area for formula extraction and recognition.

Formula recognition has gained research importance in recent years. In the past few decades, many researchers have developed a promising number of approaches for mathematical document recognition [1-11]. But, most works we survey focus on mathematical formulas themselves and do not recognize the whole mathematical

document. They assume that the regions containing mathematical formulas are already known. OKAMOTO and MIYAZAWA [11] note that in their tests, table and picture areas were excluded and the distinction between text lines and mathematical expressions was specified manually. Additionally, most papers delve into recognizing two-dimensional mathematical expressions, without being specific, can not handle all kind of formulas. They generally recognize simple equations but not matrix or system of equations. This paper describes current results of a system that separates mathematical formulas from ordinary text on a scanned page of mixed material. We explore the extend to which this separation can be automated in the context of printed mathematical documents. Our aim is to start from digitally scanned images of documents containing mathematical formulas and to extract them in order to not disturb the OCR application not yet trained for formula recognition and restructuring. Such a tool could be really useful to be able to recognize mathematical documents and re-use them in other applications.

In this paper, we will provide in section 2, a survey of existing work in mathematical formula extraction. Besides, we will describe in section 3, our proposal approach which is then detailed in sections 4 and 5. Afterwards, we will discuss some experimental results in section 6 and 7. We will close the paper with some conclusions and prospects.

2. State of art

So far, to the best of our knowledge, papers that provide literature survey of the area of mathematical formula extraction research are very rare. A paper by LEE and WANG [12] is directed to our task, but uses somewhat different techniques. They present a system for extracting both isolated and embedded mathematical expressions in a text document. Text lines are labelled as isolated expressions based both on internal properties and on having increased white space above and below them. There are good first-cut heuristics but make

mistakes : titles are often labelled as isolated formulas. The remaining text lines consist of a mixture of pure text and text with embedded expressions. They treat embedded expressions, by first recognising the characters. Characters that are known to be mathematical are used as seeds for growing geometric “trees” of mathematical expressions, heuristically attaching symbols that are adjacent including those in super or subscripting or matrix structures. The embedded mathematical expressions are then extracted from text based on some primitive tokens. The system determines whether a primitive token belongs to an embedded mathematical expression according to some basic expressions forms. The major errors are due to similar symbols. The authors not attempt to confirm that the localised sections contain mathematical expressions, leaving a parser and future corrective procedures for future work.

To find mathematical expressions on a scanned page, FATEMAN [13] proposes a system that identifies all connected components by observing character size and their font. Based on such identification, the system separates all items into two bags : math and text. The text bag includes all Roman letters, Italic numbers. The math bag includes punctuation, special symbols, Italics letters, Roman digits, and other marks (horizontal lines, dots), etc. The math bag components are then grouped into zones according to their proximity. But some components such as dots, commas and parentheses, that correctly belong in text, might be absorbed in a math zone. After this grouping within math bag, some symbols (isolated dots, commas, and parentheses, etc.) could remain isolated. These symbols might be attributed either to math or text given appropriate context. If they appear to be too far from other math symbols to be grouped together with them, they will be moved to text bag. Isolated italic letters or isolated Greek letters remain as math. Next, the system joins up the text bag into groups according to their proximity. Some text words

which are relatively isolated from other text, but are within zones that have been previously established in the math bag, will be moved into the math bag. Finally, the bags of math vs. text, must be reviewed and corrected. With this method, italic words will generally be recognised as mathematical expressions although they may be either mathematical expressions or text. The strings of numbers and symbols are considered as mathematical expressions although sometimes appear in text. Figures or line-drawings that include dotted or dashed lines may contain connected components that look like mathematical expressions. The problem of separating these out and treating them like figures is not yet solved.

INOUE and al [14] describe a new system of OCR, which can handle Japanese scientific documents. After the extraction of text lines, including mathematical formulas from a scanned page image, the system segments each line into Japanese text area and mathematical formula area. The Japanese area contains only Japanese characters, while the mathematical area covers the complement. The segmentation and the recognition of the Japanese characters are done at the same time using an adapted OCR (for kanji, kana and Japanese punctuation symbols). Even though the OCR always gives correct results, the segmentation remains a simplest task considering as mathematical what is not recognised by the OCR, which is not always true.

To separate the mathematical text from plain text, TOUMIT et al. [15] recently proposed another approach based on a physical and a logical segmentation methods. Physical segmentation is achieved to extract the document layout such as blocks, lines, characters and words. Logical segmentation consists in formula detection by following two steps: 1) detection of “big formulas” considering their centred position in the page and the lack of

abandon text, 2) location of “small formulas” in the text lines by finding special symbols such as =, +, <, > and specific context extension from these symbols.

This paper addresses the issue of locating mathematical formulas in paper documents for both cases: isolated formulas and embedded formulas in the text lines. The segmentation processes are performed directly on the document image without using any OCR system. The main reasons are : 1) current OCRs are not capable to furnish acceptable segmentation rate on mathematical documents because they are more adapted to linear writing (text lines) than to two-dimensional writing, 2) the embedded formulas within the text create hostile context for general recognition by OCR manifested by formulas compression leading to different font variation, 3) OCRs are incapable to produce the exact structure of the formulas.

Another aim of the propose approach is to accompany OCR (which is generic in nature) on uncommon part of the content obtained by a generic segmentation approach: extraction of specific symbols, extension of the context around these symbols and segmentation of the material containing these contexts. The need of such methodology can be formulated for different heterogeneous documents like mechanical, chemical and geographical documents.

3. Propose approach

It is obvious that separating mixed materials should help the accuracy of commercial OCR programs. We propose to improve the OCR success rate on mixed material by separating mathematical expressions from the usual text. To find where formulas are located on the document, a top down approach (global to local segmentation) is performed. The extraction of isolated formulas is less complex in principle because there is some helpful information like vertical spaces which identify the mathematical expressions

directly. They have distinctive lower density compared to a normal text paragraph and unusual line statistics. Embedded formulas are extracted, by location of their most significant mathematical operators, then extension to adjoining operands and operators using contextual rules until delimitation of the whole formulas spaces.

The system performs the following main tasks. First, the document is scanned, its image is straightened and its connected components are extracted. Each of extracted connected components is associated with a bounding box. Using the attributes deduced from the coordinate of the bounding boxes, the system assigns a label to each of them according to the role it can play in formula composition. This primary labeling allows a global segmentation of the document by extraction of lines and their classification into lines of text or lines of isolated formulas. For embedded formulas, a local segmentation of text lines is necessary. It needs a finer labeling to locate some mathematical operators. All the characters and operators, when grouped properly, allow to embedded formulas to be separated from usual text. However, proper grouping of operators in mathematical formula is not trivial. Firstly, there are many operators. Each of them has its own grouping criteria. Secondly, there are two types of operators, namely, explicit and implicit operators. Explicit operators are operators symbols (Σ , Π , \int , $=$, $+$, etc.) while implicit operators are spatial operators (subscripts and superscripts). Thirdly, some symbols may represent different meaning in different contexts. These properties together make the extraction process very difficult even when all characters and symbols can be recognized correctly. An overview of the entire system is given in Figure 1.

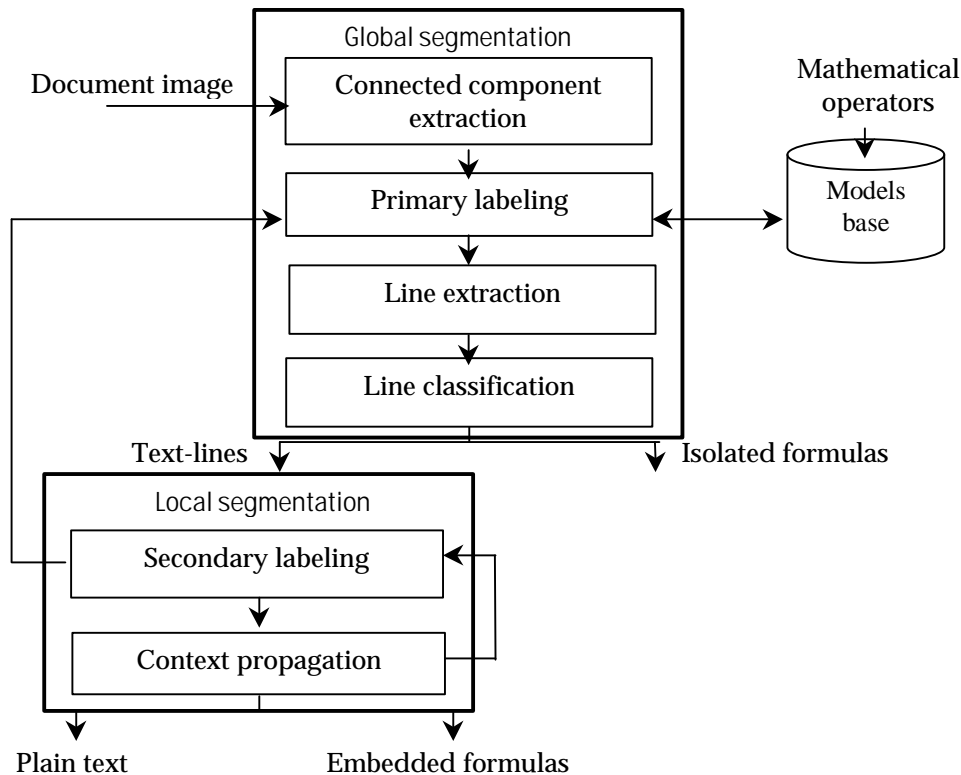


Figure1: System overview

The processing levels as they are shown, will be detailed in the following sections.

4. Global segmentation

The main goal of the global segmentation is to identify particularly isolated formulas. This is based on symbol extraction, detection of lines containing these symbols and consecutive line merging for fractions. These procedures are detailed in the following sections.

4.1. Connected component extraction

In many cases, the extracted connected components correspond to the characters on the page image, although they can be both character fragments or merged characters. Each connected component (noted c) is described by the co-ordinates of the upper left (X_{min} , Y_{min}) and the lower right (X_{max} , Y_{max}) corners of its bounding box and the number of its black pixels (nbp) (See Figure 2).

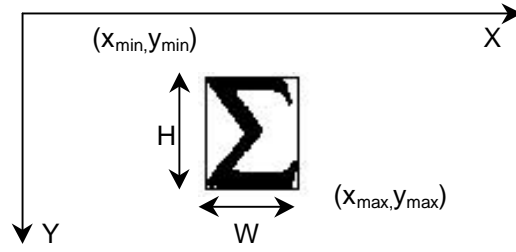


Figure 2 : The bounding box of a connected component

Afterwards, the connected component filtering is taken to discard noise, some diacritical and punctuation signs, large graphics, vertical and horizontal separators since they could not be parts of mathematical formulas.

4.2. Features and spatial relations

Let $W(c)$ and $H(c)$ be respectively the width and the high of the bounding box of c . The aspect ratio R , the area A and the density D of each connected component c is computed as follows : $R(c)=W(c)/H(c)$, $A(c)=W(c)*H(c)$ and $D(c)=nbp/A(c)$.

The relative size $X(c_l, c_r)$ and position $Y(c_l, c_r)$ of a pair of connected components (c_l : the left component and c_r .the right component) are determined as follows (See Figure 3).

$$X(c_l, c_r)=H(c_r)/H(c_l) \text{ and } Y(c_l, c_r)=(Y_{max}(c_l)-Y_{min}(c_r))/H(c_l).$$

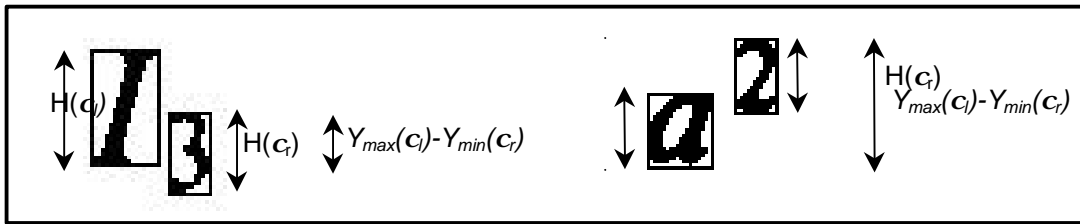


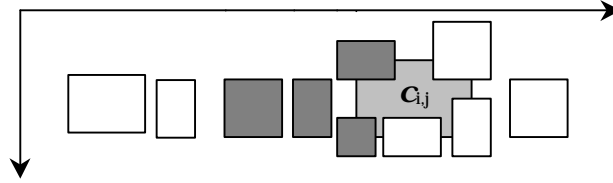
Figure 3 : The relative size and position of a pair of connected components

Let $c_{i,j}$ be the i^{th} connected component belonging to the j^{th} line L_j of the document image. The connected component of the same line are sorted by ascending order of their X_{min} . $nc(L_j)$ is the number of the connected components in L_j . Let $D(c_{i,j}, c_{i-1,j})=X_{min}(c_{i,j})-X_{max}(c_{i-1,j})$ be the distance between two consecutive connected components. We define the spatial relations between a pair of connected components as follows :

- LN (Left Neighbourhood) : The list of the connected components in the left vicinity of $c_{i,j}$.

$$LN(c_{i,j}) = [c_{k,j} \text{ such as } i-1 \leq k \leq i \text{ and } X_{min}(c_{k,j}) \leq X_{min}(c_{i,j})] \cup [c_{p,j} \text{ such as } 1 < p \leq i-1 \text{ and}$$

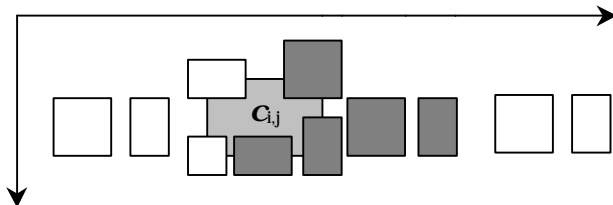
$$D(c_{p+1,j}, c_{p,j}) = D(c_{p,j}, c_{p-1,j})].$$



- RN (Right Neighbourhood) : The list of all the connected components in the right vicinity of $c_{i,j}$.

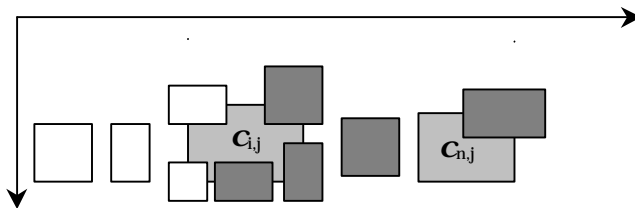
$$RN(c_{i,j}) = [c_{k,j} \text{ such as } i+1 \leq k \leq nc(L_j) \text{ and } X_{min}(c_{k,j}) \leq X_{max}(c_{i,j})] \cup [c_{p,j} \text{ such}$$

$$\text{as } k+1 \leq p < nc(L_j) \text{ and } D(c_{p,j}, c_{p-1,j}) = D(c_{p+1,j}, c_{p,j})]$$



- DLM (Delimitation) : The list of all the connected components enclosed inside $c_{i,j}$ and $c_{n,j}$.

$$DLM(c_{i,j}, c_{n,j}) = [c_{k,j} \text{ such as } i+1 \leq k \leq n-1 \text{ and } X_{min}(c_{k,j}) \leq X_{max}(c_{n,j})]$$



4.3. Primary labeling

Mathematical formulas are represented with various kind of entities. Such entities include all possible alphabetic characters (English, Greek, Hebrew, etc.), numerals (1,2,3, etc.), math operators (+, *, -, Σ , Π , \int , (, [, etc.) and so on. Though extraction of such objects is the first step to locate mathematical formulas. Some special mathematical operators are useful to locate formulas. The most obvious example is the “equal” symbol, which appears

in many formulas. Other useful characters are the symbols of summations, products, integrals, roots, fraction bars, brackets and parenthesis, etc. To proceed with our system, we must tentatively identify many of the connected components as particular characters. Characters that are known to be mathematical (such as Σ , Π , \int , etc.) are used as tokens for formula extraction. While we do not expect 100% separation, with some training (especially on the character set in use for mathematical expressions) we expect that only a modest amount of text will be confused with mathematics.

To learn mathematical symbols, the system must analyse the large number of symbols extracted from different mathematical documents. For each instance of symbol, values of the aspect ratio, area and density are computed, observed and only the lower and the upper bounds are considered. We have study 1182 instances of mathematical symbols: 263 summation and product symbols (*SP*), 83 integrals (*IS*), 101 roots (*RS*), 109 horizontal fraction bars (*HFB*), 177 great delimiters (*GD* great brackets and parenthesis), 205 small delimiters (*SD* usual brackets and parenthesis) and 244 operators composed of small horizontal lines such as equal and subtraction signs (*OP*).

This difference in the sample size reflects the use frequency of those symbols in mathematical documents. We do not consider other operators such as '+', '*', '/', '<', '>' in the class of operators because they are often confused with some alphanumeric characters.

Let $P=\{R, A, D\}$ be the set of parameters used for mathematical symbol classification, $MS=\{SP, IS, RS, HFB, VD, SD, OP\}$, the set of labels assigned to mathematical symbols, $TS(MS)$ the training sample size of an element of MS and $LB_P(MS)$ and $UB_P(MS)$ are respectively the lower and the upper bounds of a MS element according to a parameter P .

- $LB_P(MS) = \text{Min}(P(MS_i))_{i=1, \dots, TS(MS)}$.
- $UB_P(MS) = \text{Max}(P(MS_i))_{i=1, \dots, TS(MS)}$.

In the Table 1, we give the obtained results of the training step.

Table 1. Training results

MS	TS(MS)	R(MS _i) _{i=1,...,TS(MS)}		A(MS _i) _{i=1,...,TS(MS)}		D(MS _i) _{i=1,...,TS(MS)}	
		LB _R (MS)	UB _R (MS)	LB _A (MS)	UB _A (MS)	LB _D (MS)	UB _D (MS)
SP	263	0,30	1,57	78	841	0.24	0.53
RS	101	0,57	8,96	303	10000	0.05	0.21
IS	83	0,17	0,79	136	1852	0.10	0.29
BFH	109	8,09	100	27	1317	0.16	0.99
GD	177	0,07	0,31	92	1094	0.15	0.65
SD	205	0,08	0,47	24	205	0.22	0.92
OP	244	3,18	14,75	5	28	0.53	1.00

The values of ratios and area of mathematical symbols have been normalised respectively according to the greatest ratio and the largest area.

Given the training results, we might be to assign a label to each component according to the role it could play in formula composition. The first idea to label a component consisted in computing the intersection of symbols sets such as their value parameter intervals includes those of the component. However some ambiguities could be observed when the intersection result leads to more than one label. An example of labeling a great bracket is shown in the Table 2. Two possible labels : IS and GD are provided by the system.

Table 2. Results of the primary labeling of a great parenthesis

R(χ)	A(χ)	D(χ)	MS _R (χ)	MS _A (χ)	MS _D (χ)	MS(χ)
0,19	375	0,20	{IS, GD, SD}	{SP, RS, IS, HFB, GD}	{RS, IS, HFB, GD}	{IS, GD}

To remove such ambiguities, we have dropped the idea of this binary labeling on behalf of a labeling based on the fuzzy logic. The idea is not to keep only the lower and the upper bounds of the ratio, area and density values for each type of symbol but the whole measures and construct their corresponding histograms (see an example of a histogram in Figure 4). The histogram abscissa refers to the different value classes that is the set of measures shared on regular intervals. The ordinate is the relative frequency that is the number of measures belonging to a value class divided by the total number of measures.

The ordinate could be considered as the membership degrees to the different classes of mathematical symbols [16 -20].

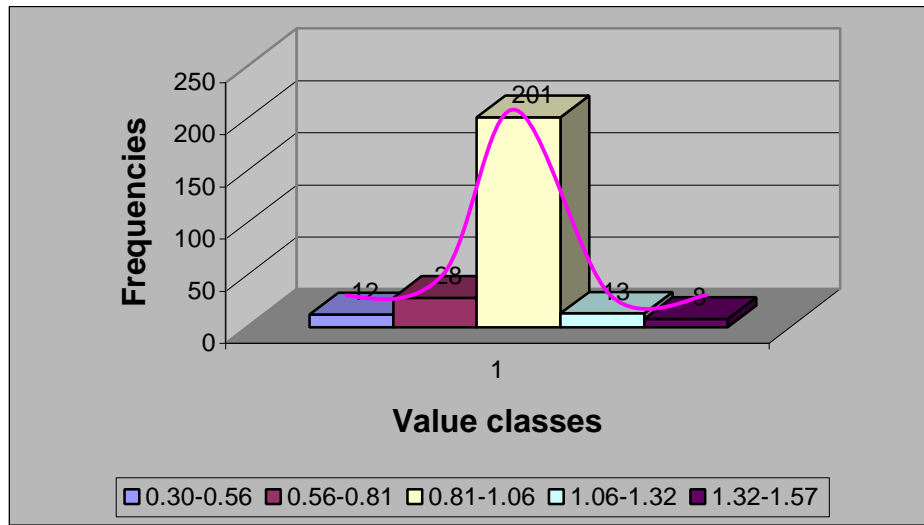


Figure 4: The aspect ratio histogram of summation and product symbols

To identify a mathematical symbol given its connected component (c), values of each parameters $P=\{R, A, D\}$ are computed. By referring to the histograms of each type of symbols, we each time keep the membership degree of that c to a type of symbols according to one parameter noted $m_{MS,P}(c)$. We then keep, for each type of symbols, the minimal membership degree of that c according to its aspect ratio, area and density (conjunction of parameters). We finally take the maximal value (disjunction of symbol types). Thus, the membership degree of that c to a class of symbol is defined as follows :

$$\begin{aligned}
 m_{MS}(c) &= \text{Max}(\text{Min}(m_{MS,R}(c), m_{MS,A}(c), m_{MS,D}(c))) \\
 &= \text{Max}(m_{SP}(c), m_S(c), m_{RS}(c), m_{HFB}(c), m_{GD}(c), m_{SD}(c), m_{OP}(c))
 \end{aligned}$$

In Table 3, we present the results obtained after a fuzzy labeling of the great parenthesis, not identified by the previous binary labeling.

Table 3. Labeling of the great bracket using fuzzy logic

MS	$\mu_{MS,R}(\chi)$	$\mu_{MS,A}(\chi)$	$\mu_{MS,D}(\chi)$	$\mu_{MS}(\chi)$	
SP	0	0	0.33	0	
IS	0	0.03	0.82	0	
RS	0.02	0.14	0.47	0.02	
HFB	0	0.05	0.14	0	
VD	0.44	0.47	0.55	0.55	
SD	0.16	0	0	0	
OP	0	0	0	0	
				$m_{MS}(\chi)$	0.55
				MS (χ)	GD

It is clear that the membership degree of that great parenthesis to class of small delimiter (0.55) is greater than the membership degree to class of integral symbol (0.02).

The fuzzy logic has been shown to be useful not only to express the non uniformity of measure distribution in classes of mathematical symbols but also to remove certain ambiguities as shown in the previous example.

A test database composed of 460 mathematical symbols: 110 SP, 45 IS, 12 RS, 56 HFB, 93 GD, 104 SD and 40 OP is used to compute the average rate of the primary labeling. About 95.3% of the connected components are well labelled (see Table 4).

Table 4. Confusion matrix of the primary labeling

	SP	RS	IS	HFB	GD	SD	OP	Not labelled
SP	100%	0%	0%	0%	0%	0%	0%	0%
RS	0%	84%	0%	0%	0%	0%	0%	16%
IS	0%	0%	100%	0%	0%	0%	0%	0%
HFB	0%	0%	0%	92%	0%	0%	3%	5%
GD	0%	0%	2%	0%	96%	0%	0%	2%
SD	1%	0%	2%	0%	2%	95%	0%	0%
OP	0%	0%	0%	0%	0%	0%	100%	0%

The most errors are due to some similar symbols. We will show how it will be possible to distinguish between them at a secondary labeling step.

4.4. Line extraction

Once the connected components are labeled, the adjacent ones are grouped into the same line following the next steps. First, the connected components are sorted by their ascending Y_{min} . Initially, the Y_{min} and the Y_{max} of the line correspond to those the first meted component. Then the line ordinates are updated by checking overlapped components. Those belonging to the same line are then sorted by their ascending X_{min} to determine the X_{min} and the X_{max} of the line (see Figure 5).

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1$$

Figure 5 : Example of line extraction

Sometimes, a line fusion step seems to be necessary especially for formula spread over than one line such as fractional, summation, product or integral expressions (see Figure 6). In such cases, the connected components of the numerators and denominators should be joined to their corresponding fraction bars. By the same way, the lower and the upper limits must be connected to symbols of summation product or integral.

①

$$\frac{a}{\sin A} = \frac{b}{\sin B} = \frac{c}{\sin C}$$

On vérifie bien que $\sum_{i=1}^6 P(E_i) = 1$, puisque $\bigcup_{i=1}^6 \{E_i\} = \Omega$ où Ω est

On vérifie bien que $\sum_{i=1}^6 P(E_i) = 1$, puisque $\bigcup_{i=1}^6 \{E_i\} = \Omega$ où Ω est

$$y_j = \frac{\sum_{i=1}^N a_{ij} x_i p(x_i \in C_j)}{\sum_{i=1}^N a_{ij} p(x_i \in C_j)}$$

Figure 6 : Examples of line fusion

In case of Figure 6, the fusion of numerator and denominator lines is not performed because of the fraction bar which overlaps with the upper limit of the summation in the denominator.

c_{ij} corresponds to the i^{th} c of the j^{th} line. It is characterized by its spatial co-ordinates, its label $MS(c_{i,j})$ and its membership degree to the class of MS noted, $\mu_{MS}(c_{i,j})$.

Once lines are extracted, isolated formulas could be located. In fact, isolated formulas are big formulas, which constitute a single line with or without very small text, and they are often centred in the page. Extracting them is an easy and quick task. We perform it using two assumptions based on their morphology (ratio) and position on the image (according to the left and the right margin). Let L_j be the j^{th} line of the image, $R(L_j)$ its aspect ratio, $nc(L_j)$ the number of its connected components, LM and RM are respectively the left and the right margin of the image. $LM = \min(X_{min}(L_k))_{k=1..nl}$ while $RM = \max(X_{max}(L_k))_{k=1..nl}$ where nl is the number of lines in the image. ISF and TXT are the two labels assigned respectively to isolated formula line and to a text line. The line classification is proceeded as follows :

If $(r_1 \leq R(L_j) \leq r_2 \text{ and } X_{max}(L_j) > (LM + RM) / 2)$ /* a quite high and long line */
 or $(d_1 \leq (X_{min}(L_j) - LM) / (RM - X_{max}(L_j)) \leq d_2)$ /* centred line */
 Then $L_j = ISF$ else $L_j = TXT$

Isolated formulas could be now extracted which restrict next processing to extraction of formulas embedded into text lines.

5. Local segmentation

Using the previous labeling, the system will try to separate embedded formulas from usual text. Mathematical text is not limited to formulas. Isolated characters in the text may

represent mathematical concepts or variables. Although these characters are not plain text since they do not obey to the grammar of standard text, we decide not extract them since an OCR program is likely to correctly read characters typeset on the normal text base-line. This notation could be extended to Greek letters and sequence of symbols. Remember that our objective is to delimit the parts of text, which could disturb OCR application.

A secondary labeling is applied. It is a finer labeling of the connected components, belonging to the same text line, where their position according to the central band of line is considered to solve some ambiguities observed at their primary labeling. In fact, with this consideration, summation, product and integral symbols could be distinguished from alphanumeric characters and oblique fraction bars, since integral, summation and product symbols are overflowing while alphanumeric characters and oblique fractions bars are not. Additionally, subscripts and superscripts which are implicit mathematical operators, could be detected since they are generally deepen or high (See Figure 7).

5.1. Secondary labeling

Six categories of components are proposed based on their topographies (position according to the central band of the line to which they belong) as shown in Figure 7.

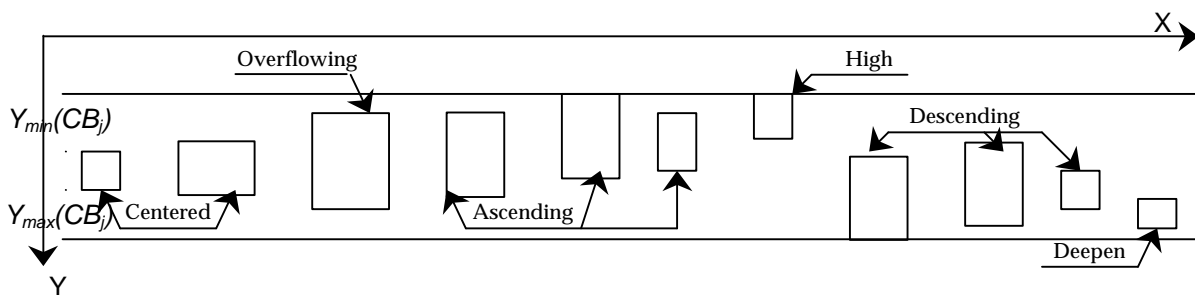


Figure 7 : Topographical classification of the connected components.

The central band ordinates : $Y_{min}(CB_j)$ and $Y_{max}(CB_j)$ of L_j , correspond respectively to the maximal horizontal projection values of Y_{min} and Y_{max} of all the connected components belonging to L_j .

Let $T(c_{i,j})$ be the topography class of $c_{i,j}$, CB_j the central band of the L_j and alh_j the average local height of L_j the given as follows :

$$alh_j = \frac{\sum_i^{nc(L_j)} Y_{\max}(c_{i,j}) - Y_{\min}(c_{i,j})}{nc(L_j)}$$

- $T(c_{i,j}) = \text{Overflowing}$ if $Y_{\min}(c_{i,j}) < Y_{\min}(c_j) - alh_j$ and $Y_{\max}(c_{i,j}) > Y_{\max}(CB_j) + alh_j$.
- $T(c_{i,j}) = \text{Ascending}$ if $Y_{\min}(c_{i,j}) < Y_{\min}(CB_j) - alh_j$ and $Y_{\max}(c_{i,j}) \leq Y_{\max}(CB_j) + alh_j$.
- $T(c_{i,j}) = \text{Descending}$ if $Y_{\min}(c_{i,j}) \geq Y_{\min}(CB_j) - alh_j$ and $Y_{\max}(c_{i,j}) > Y_{\max}(CB_j) + alh_j$.
- $T(c_{i,j}) = \text{Centred}$ if $Y_{\min}(c_{i,j}) \geq Y_{\min}(CB_j) - alh_j$ and $Y_{\max}(c_{i,j}) \leq Y_{\max}(CB_j) + alh_j$.
- $T(c_{i,j}) = \text{High}$ if $Y_{\max}(c_{i,j}) \leq (Y_{\min}(CB_j) + Y_{\max}(CB_j))/2$.
- $T(c_{i,j}) = \text{Deepen}$ if $Y_{\min}(c_{i,j}) \geq (Y_{\min}(CB_j) + Y_{\max}(CB_j))/2$.

The major errors of the labeling step are due to the ambiguities between characters such as 'l', 't', '1' and small delimiters (brackets and parenthesis) since they have similar ratio, area and density and both of them have ascending components according to the central band of the line to which they belong. These errors can be reduced using a threshold membership degree to the class of small delimiters.

As subscripts could be descending (not too deepen) and the superscripts could be ascending (not too high) and since both of them are implicit operators which are indicated by the relative location of their operands, two other features are considered to be able to detect them : the relative size X and the relative position Y (See 4.2). The obtained training results of the subscripts and superscripts are mentioned in Table 5. Let $IO = \{SUB, SUP\}$ be the set of implicit operators, $F = \{X, Y\}$ be the set of features for IO , $LB_F(IO) = \text{Min}(F(IO_i))_{i=1, \dots, TS(IO)}$ whereas $UB_F(IO_i)_{i=1, \dots, TS(IO)}$. $TS(IO)$ is the training sample size for IO .

Table 5. Training results of subscripts and superscripts

IO	TS(IO)	LB _x (IO)	UB _x (IO)	LB _y (IO)	UB _y (IO)
SUB	100	0.11	1.26	-0.21	0.75
SUP	100	0.21	1.28	1.03	2.82

To demonstrate the contribution made by the secondary labeling of the connected components to improve results of their primary labeling, an illustrative example is given in Fig. 8 and Table 7. The not labelled connected components correspond to usual characters (noted C in Table 7). $MS_1(c_{i,j})$ and $MS_2(c_{i,j})$ are the two first labels provided to $c_{i,j}$ by the system. $m_{MS_1}(c_{i,j})$ and $m_{MS_2}(c_{i,j})$ are respectively the membership degrees to MS_1 and MS_2 . $MO(c_{i,j})$ and $m_{MO}(c_{i,j})$ are respectively the label assigned to $c_{i,j}$ and its membership degree to the class of mathematical operators after a secondary labeling step. The first column in Table 6 refers to the origin identity of $c_{i,j}$. It is used to be compared with the labeling results.


Here $P(C_k)$ and  are the a priori and conditional probabilities

Figure 8. Example of embedded formula

Table 6. The labeling results of the embedded formula shown in Figure 8

$\chi_{i,j}$	$MS_1(\chi_{i,j})$	$MS_2(\chi_{i,j})$	$m_{MS_1}(\chi_{i,j})$	$m_{MS_2}(\chi_{i,j})$	$T(\chi_{i,j})$	$IO(\chi_{i,j})$	$m_{IO}(\chi_{i,j})$	$MO(\chi_{i,j})$	$m_{MO}(\chi_{i,j})$
C	SP	C	0.22		Descendante	ID	0.04		
SD	SD	GD	0.35	0.02	Ascendante	EX	0.37	PD	0.35
C	C	C			Centrée				
SD	C	C			Ascendante	EX	0.39		
C	SP	C	0.30		Ascendante				
SUB	C	C			Descendante	ID	0.51	ID	0.51
SD	PD	C	0.35		Ascendante	EX	0.08	PD	0.35

For subscripts, superscripts, summation and product symbols and small delimiters, only ones having a membership degree greater than a threshold value (0.5 for SUB and SUP, 0.3 for SP and 0.2 for SD) are retained.

5.2. Context propagation

Before we can interpret the identified operators, we must first group them properly into units. Proper combination of operators must be syntactically correct in a mathematical sense. This can be done by some conventions in writing mathematical formulas as heuristics. For summation, product and integral symbols and operators, the propagation of context is done around them. For parenthesis, brackets and roots, it is done between them. For fraction bars, it is done above and under them. That leads to the detection of sub-expression. Let $MF_{i,j}$ be the i^{th} mathematical formula of the j^{th} line. The next rules are used to propagate the context around the found mathematical operators.

- **R1:** The symbols “ Σ ”, “ Π ”, “ \int ” are usually accompanied by limit expressions appearing above or below the symbols. So, if a summation, product or an integral symbol are detected, then their associated limits will be connected to them in addition of their right neighbourhood (see in Figure 9).

$$if(MO(c_{i,j}) \hat{I} \{SP, RS\}) \text{ then } MF_{i,j} = [c_{i,j}] \hat{E} LN(c_{i,j}) \hat{E} RN(c_{i,j})$$

- **R2:** Each component enclosed inside a radical symbol should compose a formula (see in Figure 9).

$$if(MO(c_{i,j}) \hat{I} \{RS\}) \text{ then } MF_{i,j} = [c_{i,j}] \hat{E} LN(c_{i,j}) \hat{E} RN(c_{i,j})$$

- **R3:** Each component placed above or under a horizontal fraction bar should compose a formula (see in Figure 9).

$$if(MO(c_{i,j}) \hat{I} \{HFB\}) \text{ then } MF_{i,j} = [c_{i,j}] \hat{E} ["c_{k,j} \text{ such as } 1 \text{ and } X_{max}(c_{k,j})^3 \\ X_{min}(c_{i,j})] \hat{E} ["c_{p,j} \text{ such as } i+1 \text{ and } X_{min}(c_{p,j}) \text{ and } X_{max}(c_{i,j})]$$

- **R4** : Each component enclosed inside a pair of great delimiters should form a formula (see in Figure 9).

if($MO(c_{i,j}) \hat{I} \{GD\}$) *then* $MF_{i,j} = [c_{i,j}] \hat{E} [S_{c_{n,i}}$ such as $i+1 \hat{L} \hat{L} \hat{L} c(L_j)$ and $MO(c_{n,i}) \hat{I} \{GD\}] \hat{E} DLM(c_{i,j}, c_{n,i})$

- **R5** : If an operator or a reduced number n of characters is found inside a pair of small delimiters then all of them constitute one formula. If the components before the formula are more close to the formula than to their left neighbours, then they will be joined to the formula (see ①, , , and in Figure 9).

if($MO(c_{i,j}) \hat{I} \{SD\}$) *then* $MF_{i,j} = [c_{i,j}] \hat{E} [S_{c_{n,j}}$ such as $i+1 \hat{L} \hat{L} \hat{L} c(L_j)$ and $MO(c_{n,i}) \hat{I} \{SD\}$ and $S_{k,j}$ such as $i+1 \hat{L} \hat{L} \hat{L} -1$ and $(MO(c_{k,i}) \hat{I} \{SUB, SUP, RS, SP, HFB, OP\}$ or $k-i \hat{L})] \hat{E} DLM(c_{i,j}, c_{n,j}) \hat{E} LN(c_{i,j})$

- **R6** : If an operator is found, then its left and right operands will be joined to it (see ①, , , , and in Figure 9).

if($MO(c_{i,j}) \hat{I} \{OP\}$) *then* $MF_{i,j} = [c_{i,j}] \hat{E} LN(c_{i,j}) \hat{E} RN(c_{i,j})$

- **R7** : When a subscript or superscript is identified, it is grouped with its closest neighbour. If the later is its right neighbour and it is a subscript or a superscript then the left neighbour must be joined to the formula (see ①, , and in Figure 9).

if($MO(c_{i,j}) \hat{I} \{SUB, SUP\}$)

then if($D(c_{i,j}, c_{i-1,j}) \hat{L} D(c_{i+1,j}, c_{i,j})$)

then $MF_{i,j} = [c_{i-1,j}, c_{i,j}]$

else if($MO(c_{i+1,j}) \hat{I} \{SUB, SUP\}$)

then $MF_{i,j} = [c_{i-1,j}, c_{i,j}, c_{i+1,j}]$ *else* $MF_{i,j} = [c_{i,j}, c_{i+1,j}]$

$\textcircled{1}$ pour $x = L$, $v_i(x) = M$
 $\omega_i = \omega_j$ and $L(\omega_i, \omega_j) = 1$ for $\omega_i \neq \omega_j$
 probability vector $\mathbf{P}_i^0 = (p_{i1}^0, \dots, p_{im}^0)$
 calculated as $\mu_B = \frac{\alpha}{\alpha + \beta}$, and
 where $R(t) \equiv \int_0^t r(u) du$. By taking
 For $m = 2$ this reduces to $V_n \sim \sqrt{2/\pi} (1/\sqrt{n})$.

Figure 9 : Examples of local context analysis

- **R8** : Two adjacent or overlapped formulas constitute one formula (see Figure 10).

$if(D(MF_{i,j}, MF_{i+1,j}) \leq 0) \text{ then } MF_{i,j} = MF_{i,j} \dot{\cup} MF_{i+1,j}$

- **R9** : Two formulas, separated by a reduced number n of components (not more than 5) should compose one formula (see Figure 10).

$if(D(MF_{i,j}, MF_{n,j}) > 0 \text{ and } i-5 \leq n < i) \text{ then } MF_{i,j} = MF_{i,j} \dot{\cup} [\cup_{i < k < n} MF_{k,j}] \dot{\cup} MF_{n,j}$

Here $P(C_k)$ and $p(x|C_k)$ are the a priori and conditional probabilities, respectively. The convergence to a local minimum by the rule (7) is guaranteed when an infinite sequence of random observation $\{x\}$ are presented during training and the conditions $\sum_{t=1}^{\infty} \epsilon_t \rightarrow \infty$, $\sum_{t=1}^{\infty} \epsilon_t^2 < \infty$ are satisfied.

Figure 10 : Examples of extension of the context

6. Current results

The current developed system to extract mathematical formulas runs under PC Pentium II. A SCANJET scanner is used to scan the mathematical document and save it as a binary image file at a resolution of 300dpi. In the experiments, the system is trained using 1182 mathematical symbols, 200 implicit operators and tested by 460 symbols, 100 implicit operators, 300 formulas and a variety of mathematical documents.

To evaluate the rate of mathematical formula extraction, we have taken into account well extracted formulas and formulas which are incorrectly extracted but with penalty. We have even penalised formulas that were entirely missed by the automatic process since they exist but the system fails to extract them. Similarly, we have penalised formulas

mistakenly extracted since the system confuses some parts of the text with mathematical formulas. The penalisation coefficient varies according to the gravity of the extraction error. Indeed, formulas incorrectly extracted are few penalised because even if the system does not a precise delimitation of the formula, it succeeds to localise in major cases the parts, which disturb the OCR application. By the same way, the formulas that are mistakenly extracted are few penalised since any OCR system is capable to recognise plain text. On the other hand, the non-extracted formulas are more penalised since they exist but the system fails to detect them.

Let NF be the number of formulas that actually exist in the document, $NWEF$ the number of well extracted formulas, $NMEF$ the number of mistakenly extracted formulas, $NNEF$ the number of non extracted formulas, $NIEF$ the number of incorrectly extracted formulas and a, b, c and ϵ their respective penalisation coefficients. The extraction rate ER is computed according to the following formula :

$$ER = \frac{aNWEF + bNMEF + cNNEF + \epsilon NIEF}{aNF}$$

where a, b are positives (total and partial success) and c, ϵ are negatives (partial and total failing), $b < a$, $|\epsilon| < |c|$ and $a + |\epsilon| = b + |c| = 0,5$.

After choice of the penalisation coefficients values, the rate is given as follows:

$$ER = \frac{0.45NWEF + 0.41NMEF - 0.09NNEF - 0.05NIEF}{0.45NF}$$

The obtained results indicate that approximately 93% of formulas could be extracted from images of the mathematical documents. The time of formula extraction varies according to the number of the connected components in the document, its incline degree

as well as the type and the placement of formulas included in the document. The system takes 0.02 second by a connected component that is 1 to 2 seconds by formula.

An example of mathematical document segmentation which contains 8 lines and 27 formulas (1 isolated formula and 26 embedded formulas) is shown in Figure 11. The extraction results are as follows :

- 1300 connected components,
- 26 formulas extracted (1 isolated formula and 25 embedded formula),
- 21 formulas are well extracted,
- 5 formulas are mistakenly extracted,
- 0 formula not extracted,
- 0 formula incorrectly extracted,
- Extraction time = 30s,
- Extraction time/component=0s.02,
- Extraction time/formula=1s.14
- Extraction rate = 94.65%.

denoted θ , takes value in $\Omega = \{\omega_1, \dots, \omega_c\}$ with probabilities $\{p(\omega_1), \dots, p(\omega_c)\}$, respectively and that \mathbf{x} is a realization of a random vector \mathbf{X} characterized by a conditional distribution $p(\mathbf{x}|\theta)$, $\theta \in \Omega$. Thus, the task is to find a measurable mapping $\psi: \mathbb{R}^d \rightarrow \Omega$ such that the expected loss function $R(\psi) = E\{L(\psi(\mathbf{X}), \theta)\}$, called risk, is minimal. Here $L(\omega_i, \omega_j)$ is the loss incurred by taking action ω_i when the class is ω_j . In this paper we assume, without loss of generality, that $L(\omega_i, \omega_j) = 0$ for $\omega_i = \omega_j$ and $L(\omega_i, \omega_j) = 1$ for $\omega_i \neq \omega_j$ and then $R(\psi) = P(\psi(\mathbf{X}) \neq \theta)$ is called the probability of error. It is well known that an optimal rule ψ^* (the Bayes rule) which minimizes $R(\psi)$ is of the following form $\psi^*(\mathbf{x}) = \arg \max_{1 \leq i \leq c} p_i(\mathbf{x})$, where $p_i(\mathbf{x}) = P(\theta = \omega_i | \mathbf{X} = \mathbf{x})$, $i = 1, \dots, c$ are the posteriori probabilities. Let R^* denote the Bayes risk, i.e., the risk of the Bayes rule. In practice we rarely have any information about the distribution of the pair (θ, \mathbf{X}) , instead there is in our disposal a training set $\eta_n = \{(\theta_1, \mathbf{X}_1), \dots, (\theta_n, \mathbf{X}_n)\}$, i.e., a sequence of pairs (θ, \mathbf{X}) distributed like (θ, \mathbf{X}) , where \mathbf{X}_j is the feature vector and θ_j is its class assignment. An empirical classification rule ψ_n is a measurable function of \mathbf{X} and η_n . It is natural to construct a rule which resembles the Bayes rule, i.e., by replacing $p_i(\mathbf{x})$ by its estimate $\hat{p}_i(\mathbf{x})$. A popular nonparametric classification technique is the kernel classifier being defined as follows

$$\psi_n(\mathbf{x}) = \arg \max_{1 \leq i \leq c} \sum_{j=1}^n \mathbf{1}(\theta_j = \omega_i) W\left(\frac{\mathbf{x} - \mathbf{X}_j}{b}\right), \quad (1.1)$$

Figure 11 : Example of mathematical document extraction

7. Analysis of extraction errors

The main errors of formula extraction are due to :

- confusion of some alphanumeric characters such as 'l', 't', '1' with brackets and parenthesis (see in Figure 12),
- confusion of the letter 'f' with integral symbol (see in Figure 12),
- confusion of some hyphens with the subtraction signs (see in Figure 12),
- confusion of some diacritical and punctuation signs with subscripts and superscripts (see ①, and in Figure 12),
- Some subscripts and superscripts are not detected while some characters are incorrectly labelled as subscripts or superscripts (see in Figure 12). That is had to the image incline or degradation or document typography.

- Confusion of some text lines with isolated formulas because they are typeset so tightly that descenders from one text line very nearly touch ascenders or superscripts from the line below (see in Figure 12).

Consequently, some parts of the text are considered as mathematical formulas while some formulas are mistakenly extracted or even they are not extracted.

① qui correspond à
neural networks,
et est appelée intégrale de f
weighted least-squares estimation.
we see that the best x_3 is
lows, we drop subscript i from $f_i(\mathbf{x})$

Theorem 2, (2.18), belonging to

where u_j is the centre of the local density. The sets $\{\lambda_{j1}, \dots, \lambda_{jd}\}$ and $\{V_{j1}, \dots, V_{jd}\}$ are the sets of respectively eigenvalues and eigenvectors of the local covariance matrix Σ_j , ordered by $\lambda_{jv} \leq \lambda_{ju}$ if $1 \leq u \leq v \leq d$. P is the selected number of principal eigenvalues. The model remains thus very

Figure 12 : the major extraction errors

To improve results of extraction notably for commas, hyphens frequently considered respectively as subscripts and subtraction signs, we have used the next rules :

- **R10** : If an alone subscript is detected after a small delimiter then, it will be considered as a comma (see ① in Figure 13).
- **R11** : If an operator is found at the end of a line then, it will be considered as a hyphen rather than a subtraction sign (see in Figure 13).

For diacritical signs, generally labelled superscripts, it seems that it is necessary to recourse to the linguistic information.

It is obvious that the system could not extract correctly an embedded formula if it can not derive good labeling results, which are dependent upon typesetting.

① Let \mathbf{B} be a set of objects $\{b_1, \dots, b_n\}$, and Λ be a set of labels

$$\boxed{f_+ - t_\alpha \sqrt{\frac{f_+(1-f_+)}{n}} < p < f_+ + t_\alpha \sqrt{\frac{f_+(1-f_+)}{n}}} \text{ avec } t \text{ corres-}$$

Figure 13 : Correction of some error extraction

8. Conclusion and future works

By providing the required value-added to scanned mathematical documents images, we aimed to support higher level tasks such as the automatic extraction of mathematical formulas. Work on mathematical formulas may ultimately be beneficial to a wider audience involved with digital library projects, especially those concerned with scientific document storage and access.

In this paper, we tried to report a system that extracts mathematical formulas automatically from images of printed documents in order to mask them out in the OCR process, while on the other hand being able to analyse the content of formulas. The main goal is to have an OCR free system for the separation of text versus mathematical expressions, hence it is mostly based on reasoning on bounding boxes of elements of the formulas. Our method is designed to extract formula even before knowing the identities of the symbols involved. In other words, it only uses information about the bounding boxes of symbols. One of the merits of this approach is that the characters need not be recognised by an OCR system. This method is certainly useful when the character and symbol recognition module fails. We have introduced fuzzy logic, which has been useful to identify mathematical operators and to delimit formulas by propagation of the context. Though a satisfactory rate of extraction is obtained, more research is still required to be able to attain human-like performance. Further work is required to extend this method to low quality documents with broken or touching characters. In fact, for low-resolution,

noisy, or poorly scanned images, this processing may be not so efficient. Old papers may also do not scan well even at high resolution. We plan to deal with more complex formulas and confirm efficiency and performance of our method using a large database of mathematical formulas.

References

- [1] R.H. Anderson, "Two-Dimensional Mathematical Notation", In proceedings of *Syntactic Pattern Recognition Applications*, K.S. Fu, Ed. Springer Verlag, NewYork , 1977, pp. 147-177.
- [2] A. Belaïd, and J. P. Haton, "A syntactic Approach for Handwritten Mathematical Formula Recognition", In *IEEE Trans. PAMI*, vol. 6. n°1, 1984, pp. 105-111.
- [3] A. Grbavec and D. Blostein, "Mathematical expressions Recognition Using Graph Rewriting", In proceedings of *ICDAR'93*, France, 1995, pp.417-421.
- [4] A. Grbavec and D. Blostein, *Handbook of character recognition and document image analysis*, world scientific publishing company, 1997, pp. 557-582.
- [5] H. J. Lee and M. C. Lee, "Understanding Mathematical Expression in a Printed Document", In Proceedings of *ICDAR'93*, Japan, 1993, pp. 502-505.
- [6] M. Okamoto and B. Miao, "Recognition of Mathematics by Using the Layout Structures of Symbols", In Proceedings of *ICDAR'91*, France, 1991, pp. 242-250.
- [7] H. M. Twaakyondo and M. Okamoto, "Structure Analysis and Recognition of Mathematical Expressions", In Proceedings of *ICDAR'95*, Canada, 1995, pp. 430-437.
- [8] J. Ha, R. M. Haralick, and I. T. Phillips, "Understanding mathematical expressions from document images", In Proceedings of *ICDAR'95*, Canada, 1995, pp. 956-959.
- [9] Z. X. Wang and C. Faure, "Structural analysis of handwritten mathematical expressions", In Proceedings of *ICPR'88*, Washington, 1988, pp. 32-34.

- [10] S. Lavirotte and L. Pottier, "Optical formula recognition", In Proceedings of *ICDAR'97*, Germany, 1997, pp. 357-361.
- [11] M. Okamoto and A. Miyazawa, "An experimental implementation of a document recognition system for papers containing mathematical expressions", In H.S. Baird et al(ed). *Structured Document Image Analysis* Springer-Verlag, 1992, pp. 36-51
- [12] H. J. Lee and J. S. Wang, "Design of mathematical expression recognition system", In Proceedings of *ICDAR'95*, Canada, 1995, pp.1084-1087.
- [13] R. Fateman, T. Tokuyasu, B. Berman and N. Mitchell, "Optical Character Recognition and Parsing of Typeset Mathematics", In *J. of Visual Commun. And Image Representation* vol 7 no. 1, March 1996, pp. 2-15.
- [14] K. Inoue R. Miyazaki, and M. Suzuku, "Optical recognition of printed mathematical documents", In Proceedings of *ATCM'98*, 1998.
- [15] J.-Y Toumit, S. Garcia-Salicetti, H. Emptoz, "A Hierarchical and RECURSIVE Model of Mathematical Expressions for Automatic Reading of Mathematical Documents", In Proceedings of *ICDAR'99*, India, 1999, pp. 116-122.
- [16] A. Kacem, A. Belaïd, and M. Ben Ahmed, "EXTRAFOR : Automatic EXTRACTION of mathematical FORMulas", In Proceedings of *ICDAR'99*, Bangalore-India, 1999, pp. 527-530.
- [17] A. Kacem, A. Belaïd, and M. Ben Ahmed, "Extraction Automatique de Formules à partir d'images de Documents Scientifiques", In Proceedings of *RFIA'00*, Paris-France, 2000.
- [18] A. Kacem, A. Belaïd, and M. Ben Ahmed, "Automatic Segmentation of Mathematical Documents", In Proceedings of *ACIDCA'00*, Monastir-Tunisia, pp. 86-91,2000.

- [19] A. Kacem, A. Belaïd, and M. Ben Ahmed, “Embedded Formula Extraction”, In Proceedings of *ICPR'00*, Barcelone-Espagne, 2000.
- [20] A. Kacem, A. Belaïd, and M. Ben Ahmed, “Extracteur de formules de documents mathématiques”, In Proceedings of *CIFED'00*, Lyon-France, pp. 295-304, 2000.