Embedded Formulas Extraction

A. KACEM
ENSI-RIADI-Tunisia
akacem@isg.rnu.tn

A. BELAID
LORIA-CNRS-France
a*belaid@loria.fr*

M. Ben AHMED
*RIADI-Tunisia*mailto:abelaid@loria.fr *mbenahmed@serst.rnrt.tn*

Abstract

A new approach for separating mathematics from usual text is presented. Contrary to the existing methods, it is more oriented toward the segmentation than the recognition, isolating the formulas outside and inside the text lines. The objective is to delimit a part of text which could disturb the OCR application, not yet trained for formula recognition and restructuring. The method is based on an adaptive segmentation working at two levels 1) A primary labelling identifies the more characteristic symbols; 2) A secondary labelling extends the context of the symbols for delimiting the formula inside the text.

Experiments done on some commonly seen mathematical documents, show that our proposed method can achieve quite satisfactory rate making mathematical formulas extraction more feasible for real-world applications. The average rate of primary labelling of mathematical symbols is about 95.3% and their secondary labelling can improve the rate about 4%. Thus, about 95% of formulas are well extracted from images of documents printed with high quality.

1. Introduction

Formulas are involved in mathematical documents, either as isolated formulas, or embedded directly into a text-line. They are in general two-dimensional structured patterns. Typically, they consist of special symbols and Greek letters in addition to Latin letters and digits. Moreover, characters and symbols may appear in various positions, possibly of different sizes. Many works have been done since the sixties on recognition of mathematical formulas [1-11]. But most of them assume that recognition system begins with isolated mathematical formulas or specified manually. Additionally, they 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 extract mathematical formulas automatically from images of printed documents. Our aim is to start from digitally scanned images of documents containing formulas and to extract them in order to not disturb the OCR application. Such a tool could be really useful to be able to recognize mathematical formulas and re-use them in other applications.

2. Previous works

Very few papers have addressed specific problems related to mathematical formulas extraction. A recent 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 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. Unfortunately, Lee and Wang do not attempt to confirm that the localised sections contain mathematics, leaving a parser and future corrective procedures for future work. Additionally, they do not take advantage of the nature of the character font information although italics could be a key feature of mathematical text as mentioned by Fateman [13].

To find mathematics on a scanned page, Fateman tried to separate mixed material into two (or more) streams, with only conventional non-math text handled by the usual OCR text-based heuristic analysis. The second stream, consisting of material judged to be mathematics, can be fed to a specialised recogniser. If that fails to decode it, it can be passed on to yet a third stream including diagrams, logos or halftones. To proceed with this system, we must generally distinguish italic from roman letters, recognise digits and identify

dots and horizontal lines. But italics words will generally be recognised as mathematics and so this may need human intervention to separate the streams of data.

Those methods use OCR systems and assume that what it is not a text is a formula. It would be pleasant to declare that one need go no further than this level to discriminate reliably between mathematical formulas and usual text. But, in our view, it is not enough especially to extract embedded formulas and as we know, conventional OCR programs have low accuracy for mathematics. This paper describes a system to separate isolated and embedded formulas, automatically and without character recognition, from ordinary text as well as from other materials.

To identify where formulas are located on the document, an adaptive segmentation is used. The idea is to do labelling at several steps : extraction of isolated then embedded formulas by location of their most significant symbols, then extension to adjoining symbols using contextual rules until delimitation of the whole formulas spaces.

This paper is organized as follows : an overview of the system is first presented. Some experimental results are then briefly presented and discussed. A conclusion and some prospects are finally given.

3. A proposal for automatic extraction of formulas

Many steps are followed for formula extraction. First, a set of connected components (CCXs) is extracted. Each of extracted CCX is associated with a bounding box. Using the attributes deduced from co-ordinate 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 labelling of CCXs 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 labelling of CCXs and their contextual analysis in a way to delimit formulas and separate them from usual text. In this paper, more attention is focused on embedded formulas because of the difficulties of their extraction.

4. Extraction of isolated formulas

First, CCXs are extracted from the document image. Each CCX is described by co-ordinates of upper left (X_{min}, Y_{min}) and lower right (X_{max}, Y_{max}) corners of its bounding box and the number of its black pixels (NBP). From this information, it is possible to compute its aspect ratio (R=W/H), area (A=W*H) and density (D=NBP/A) where W, H are respectively the width and high of the CCX.

After CCX extraction, it is convenient to reduce the working set of CCXs to one, which contains a higher percentage of characters and mathematical symbols. In fact, noise, diacritical and punctuation signs, large graphics, vertical and horizontal separators are discarded in order to improve both accuracy and processing speed of formulas extraction. The filtering approach is taken with the CCX aspect ratio and area attributes.

At the first labelling step, a label is assigned to each CCX according to the role it could play in formula composition. The mathematical formula (MF) is considered as a set of operands and explicit or implicit operators. Explicit operators are represented by mathematical symbols (MS) such as Functional Symbols (sum and product signs), Integral (IS) and Radical Signs (RS), Horizontal Fraction Bars (HFB), Small and Vertical Great Delimiters (SD, VGD), Binary Operators (BO), whereas implicit operators (IP) are indicated by the relative location of their operands such as subscripts (SUB) and superscripts (SUP).

To learn mathematical symbols, the system must analyse the greatest possible number of symbols deduced from different mathematical documents. For each instance of symbol, values of aspect ratios, areas and densities are computed, observed and only lower and upper bounds are considered. 1178 mathematical symbols have been studied (265 FS, 101 RS, 83 IS, 109 HFB, 171 VGD, 205 SD, 244 BO). Table 1. shows the obtained results of the training step.

Table 1. Training results

	$R(MS_i)_{i=1,...,E}$		$A(MS_i)_{i=1,...,E}$		$D(MS_i)_{i=1,...,E}$	
MS	$IB_R(MS)$	$SB_R(MS)$	$IB_A(MS)$	$SB_A(MS)$	$IB_D(MS)$	$SB_D(MS)$
FS	0.264	1.636	198	3900	0.232	0.48
RS	0.504	7.941	1435	47850	0.046	0.2
IS	0.156	0.687	660	8832	0.097	0.287
HFB	8	87.714	108	6336	0.141	1
VGD	0.056	0.260	345	4840	0.145	0.7
SD	0.06	0.414	116	990	0.216	0.927
BO	3.333	13.5	18	125	0.572	1

To remove some ambiguities when labelling CCXs, we have used fuzzy logic by introducing memberships degrees to the different classes of mathematical symbols. Those memberships degrees are deduced from histograms generated for each type of symbol [14 -15].

To identify a mathematical symbol given its CCX, values of each parameters P={R, D, A}are computed. By referring to histograms of each type of symbol, we each time keep the membership degree of that CCX to a type of symbol according to one parameter noted $\mu_{MS,P}(CCX)$. We then keep, for each type of symbol, the minimal membership degree of that CCX according to its aspect ratio, density and area. We finally take their maximal

value. Thus, the membership degree of that CCX to a class of symbol is defined as follow :
$\mu_{MS}(CCX)=Max(Min(\mu_{MS,R}(CCX),\mu_{MS,A}(CCX),\mu_{MS,D}(CCX))$ where MS={FS, IS, RS, HFB, VGD, SD, BO} $=Max(\mu_{FS}(CCX), \mu_{IS}(CCX), \mu_{RS}(CCX), \mu_{HFB}(CCX), \mu_{VGD}(CCX), \mu_{SD}(CCX), \mu_{BO}(CCX))$.

By testing our system using 460 mathematical symbols, an average rate of 95,3% is reached for the first labelling of CCXs.

After this primary labelling step, horizontal adjacent CCXs are grouped into the same line. CCXs belonging to the same line are then sorted by ascending $X_{min.}$ Once lines are extracted, isolated formulas could be located based on the height of their lines, generally superior than the average height of lines, and their page-setting. Thus, problem of isolated formulas extraction is solved which restrict next stages to embedded formulas.

5. Extraction of embedded formulas

Using the previous labelling, the system try to separate formulas from pure text. A second labelling of CCXs is applied. It is a finer labelling of CCXs, 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 labelling. In fact, with this consideration, functional and integral symbols could be distinguished from characters, digits and oblique fraction bar, since integral and functional symbols are overflowing while characters, digits and oblique fractions bars are not. Additionally, subscripts and superscripts could be detected since their CCXs are generally deepen or higher. For those having descending or ascending components, two other features are considered : the relative size : X=RS/LS (RS: Right component Size, LS: Left component Size) and the relative position : Y=D/LH (D: Distance between the top of the right component and the bottom of the left component).

Afterwards, the context of found operators is analysed and extended in order to separate mathematical material from text in document. There are some heuristics rules used for this purpose which depend on the type of mathematical operators. Let :

- Distance : $D(CCX_{i,j},CCX_{k,j})=X_{min}(CCX_{i,j})-X_{max}(CCX_{k,j})$
- Left Overlap : $LO(CCX_{i,j})=\{CCX_{k,j} / 1\le k\le i-1$ and $X_{max}(CCX_{k,j})\ge X_{min}(CCX_{i,j})\}$
- Left Adjacency : $LA(CCX_{i,j})=LO(CCXi,j)\cup\{CCX_{k-1,j}\}$
- Right Overlap : $RO(CCX_{i,j})=\{CCX_{k,j} / i+1\le k\le nc$ and $X_{min}(CCX_{k,j}) \le X_{max}(CCX_{i,j})\}$
- Right Adjacency: $RA(CCX_{i,j})=RO(CCXi,j)\cup\{CCX_{k+1,j}\}$
- Inside Enclosure : $ID(CCX_{i,j}, CCX_{n,j})= \{CCX_{k,j} / i+1\le k\le n-1$ and $X_{max}(CCX_{k,j}) \le X_{max}(CCX_{n,j})\}$

5.1. Subscripts and superscript extension

If a subscript or superscript is found, then 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 figures 1, 2, 5).

if($OP(CCX_{i,j})\in\{SUB,SUP\}$)
 then if($D(CCX_{i,j}, CCX_{i-1,j})\le D(CCX_{i+1,j}, CCX_{i,j})$)
  then $MF_{i,j}=\{CCX_{i-1,j}, CCX_{i,j}\}$
  else if(op(CCXi+1,j$\in$ {SUB, SUP})
   then $MF_{i,j}=\{CCX_{i-1,j}, CCX_{i,j}, CCX_{i+1,j}\}$
   else $MF_{i,j}=\{CCX_{i,j}, CCX_{i+1,j}\}$

5.2. Radial symbol extension

Each CCX enclosed inside a radical symbol should form a mathematical formula (see figure 2).

if($OP(CCX_{i,j})\in\{RS\}$)
 then $MF_{i,j}=\{CCX_{i,j},LO(CCX_{i,j}),RO(CCX_{i,j})\}$

5.3. Functional and integral symbol extension

If a functional or an integral symbol is found, then its lower and upper bounds in addition of the first component of its sub expression are joined to it (see figure 3).

if($OP(CCX_{i,j})\in\{FS, RS\}$)
 then $MF_{i,j}=\{CCX_{i,j},LO(CCX_{i,j}),RA(CCX_{i,j})\}$

5.4. Horizontal fraction bar extension

Each CCX placed above or under an horizontal fraction bar should compose a formula (see figure 4.).

if($OP(CCX_{i,j})\in\{HFB\}$)
 then $MF_{i,j}=\{CCX_{i,j},LO(CCX_{i,j}),RO(CCX_{i,j})\}$

5.5. Vertical great delimiter extension

Each CCX enclosed inside a pair of vertical great delimiters should form a formula (see figure 5).

if($OP(CCX_{i,j})\in\{VGD\}$)
 then $MF_{i,j}=\{CCX_{i,j}, \exists\ CCX_{n,j}$ / $i+1\le n\le nc$ and $OP(CCX_{n,j})\in\{VGD\}$, $IE(CCX_{i,j},CCX_{n,j})\}$

5.6. Small delimiter extension

If A mathematical operator or a reduced number of characters is found inside a pair of small delimiters then all of them constitute one formula. If the CCXs before the

formula are more closed to it then to their left neighbour then they are joined to the formula (see figures 1, 2, 3).

if(OP($CCX_{i,j}$)∈ {SD})
then $MF_{i,j}$={$CCX_{i,j}$, ∃ $CCX_{n,j}$ / i+1≤n≤nc and OP($CCX_{n,j}$)∈ {SD}, ∃ $CCX_{k,j}$ / i+1≤k≤n-1 and (OP(CCXn,j)∈ {SUB, SUP, RS, FS, IS, HFB, BO} or k-i≤3), IE($CCX_{i,j}$, $CCX_{n,j}$)}
∀$CCX_{k,j}$/1≤k≤i, if(D($CCX_{k,j}$,$CCX_{k-1,j}$)≤D($CCX_{k-1,j}$,$CCX_{k-2,j}$) then $MF_{i,j}$=$MF_{i,j}$∪{$CCX_{k,j}$}

5.7. Binary operator extension

If a binary operator is found, then their left and right operands are joined to it (see figures 1,2 , 3, 4, 5).
if(OP($CCX_{i,j}$)∈ {BO})
then $MF_{i,j}$={$CCX_{i,j}$,LA($CCX_{i,j}$),RA($CCX_{i,j}$)}

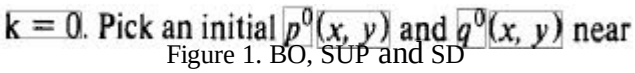

Figure 1. BO, SUP and SD

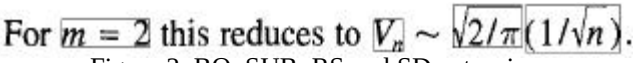

Figure 2. BO, SUB, RS and SD extension

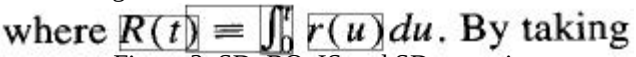

Figure 3. SD, BO, IS and SD extension

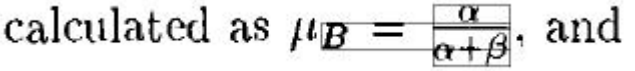

Figure 4. BO and HFB extension

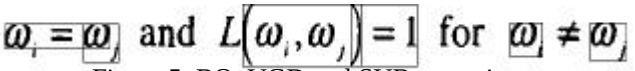

Figure 5. BO, VGD and SUB extension

5.8. Formula extension

Two horizontal adjacent or overlapped formulas constitute one formula (see figure 6).
if(D($MF_{i,j}$, $MF_{i-1,j}$) ≤ 0)
then $MF_{i,j}$ = $MF_{i,j}$∪ $MF_{i-1,j}$
Two formulas, separated by a reduced number of CCXs (not more than 5) should compose one formula (see figure 6).
if(D($MF_{i,j}$, $MF_{k,j}$) > 0 and i-5≤k<i)
then $MF_{i,j}$ = $MF_{i,j}$∪ $MF_{k,j}$

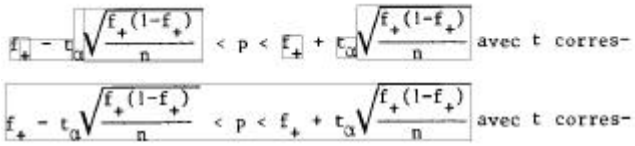

Figure 6. Formula extension

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 1178 mathematical symbols, 200 implicit operators and tested 460 symbols, 100 implicit operators and over than 200 embedded formulas. The main errors are due to confusion of subscripts and superscripts with diacritic signs (see figures 7), small delimiters with 'l' (see figure 8) and subtraction signs with the hyphens (see figure 9).

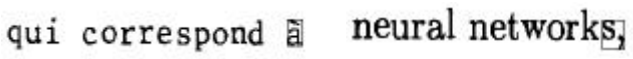

Figure 7. Confusion of a diacritic sign and punctuation signs with a superscript and subscripts

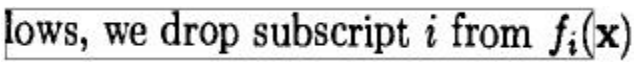

Figure 8. Confusion of 'l' with a small delimiter

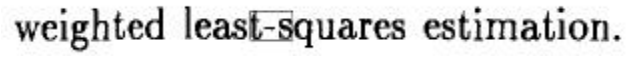

Figure 9. Confusion of subtraction sign with an hyphen

The ambiguities between diacritic, punctuation signs, subscripts and superscripts should be reduced using relative parameters which depend on the document when filtering CCXs. But, confusion of small delimiters with 'l' still remains although a membership degree threshold to the class of small delimiters is fixed to 0.20 and so this may need other parameter to distinguish them. It is obvious that the system could not extract correctly an embedded formula if it can not derive good labelling results, which are dependent upon typesetting.

7. Conclusion

By providing the required value-added to scanned 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 extract mathematical formulas automatically from images of printed documents without using OCR system. 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. This method is certainly useful when the symbol recognition module fails. We have introduced fuzzy logic at CCXs labelling which has been useful to identify symbols and consequently to delimit formulas by a

contextual analysis of their CCXs. In this coming year, we plan to deal with more complex formulas and confirm efficiency and performance of our method using a large data-base of mathematical formulas.

References

[1] R.H. Anderson, "Two-Dimensional Mathematical Notation", *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", *IEEE Trans. PAMI, vol.6. n°1*, 1984, pp. 105-111.

[3] A. Grbavec, and D. Blostein, "Mathematics Recognition Using Graph Rewriting", *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", *ICDAR'93*, Japan, 1993, pp. 502-505.

[6] M. Okamoto, and B. Miao, "Recognition of Mathematical Expressions by Using the Layout Structures of Symbols", *ICDAR'91*, France, 1991, pp. 242-250.

[7] H. M. Twaakyondo, and M. Okamoto, "Structure Analysis and Recognition of Mathematical Expressions", *ICDAR'95*, Canada, 1995, pp. 430-437.

[8] J. Ha, R. M. Haralick, and I. T. Phillips, "Understanding mathematical expressions from document images", *ICDAR'95*, Canada, 1995, pp. 956-959.

[9] Z. X. Wang, and C. Faure, "Structural analysis of handwritten mathematical expressions", *ICPR'88*, Washington, 1988, pp. 32-34.

[10] S. Lavirotte, and L. Pottier, "Optical formula recognition", *ICDAR'97*, Germany, 1997, pp. 357-361.

[11] K. Inoue, R. Miyazaki, and M. Suzuku, "Optical recognition of printed mathematical documents", *ATCM'98*, 1998, pp.

[12] H. J. Lee, J. S. Wang, "Design of mathematical expression recognition system", *ICDAR'95*, Japan,1995, pp.1084-1087.

[13] R. Fateman, T. Tokuyasu, B. Berman and N. Mitchell, "Optical Character Recognition and Parsing of Typeset Mathematics", *J. of Visual Commun. And Image Representation* vol 7 no. 1, March 1996, pp. 2-15.

[14] A. Kacem, A. Belaïd, and M. Ben Ahmed, "EXTARFOR : Automatic EXTRAction of mathematical FORmulas", *ICDAR'99*, Inde, 1999, pp. 527-530.

[15] A. Kacem, A. Belaïd, and M. Ben Ahmed, "Extraction de formules à partir de documents mathématiques", *RFIA'00*, France, 2000.