Mathematical Formula Recognition based on Modified Recursive Projection Profile Cutting and Labeling with Double Linked List

Yong-Ho Yoo and Jong-Hwan Kim

Department of Electrical Engineering, KAIST
335 Gwahangno, Yuseong-gu, Daejeon, Republic of Korea
{yhyoo,johkim}@rit.kaist.ac.kr

Abstract. Recognizing mathematical expression is important to reduce time in converting image-based documents like PDF to text-based documents that are easy to use and edit. In case of general character recognition, the sequence of character segmentation is from left to right, and from top to bottom. However, mathematical expression is a kind of two-dimension visual language. Thus, segmentation is more complex than one-dimention language. This paper proposes a modified recursive projection profile cutting method of character segmentation in images of mathematical formula, using depth first search for arranging and double linked list for re-arranging. The proposed method is demonstrated through various kinds of experiments, and shows this method can yield results of high accuracy for the recognition of mathematical formula.

Keywords: Mathematical Expression, Baseline Structure Analysis, Character recognition, Neural Network.

1 Introduction

Mathematical formula is very important in science and engineering reference because it is easily understood and expressed. Nowadays, many science references are composed of images like PDF. When we need to quote mathematical formula in PDF or modify a little, we have to convert two-dimension images into one-dimension texts via LaTeX. This job is very cumbersome as complexity of formula structures are very complex. Thus, many researchers have tried to convert two-dimension images into one-dimension texts automatically.
Over the past, there have been a number of approaches in formula recognition. There are three main issues in formula recognition. One is how to detect formula region in PDF. Xiaoyan Lin *et al.* proposed mathematical formula identification in PDF Documents [1]. They proposed to identify regions of both the isolated and embedded mathematical expressions in PDF documents. They used rule-based and learning-based method to adapt to wide range formula types. Rule-based rule proposed by them means that a line is filtered out only when it does not satisfy any of following two rules: 1) A named function appears in the line: 2)

At least one math symbol appears in the line. And proposed learning-based rule means LIBSVM, an optimized implementation of Support Vector Machine (SVM) for classifying.
Second issue is how to segment each letter in formula region that is already detected in PDF. Okamoto et al. outline a method of obtaining a structural representation of scanned images of mathematics notation using recursive projection profile cutting [2]. A projection corresponds to projecting pixels onto the x and $y-axes$ of images. The cutting process is to separate adjacent sub-expressions horizontally using a vertical projection, followed by horizontal projection to separate baselines. This process is applied recursively. However, it would happen two or more character could be recognized as one character by using this method solely.
The other is how to analysis already segmented characters and convert to one-dimension texts. Zanibbi and Blostein used seven baseline to analyze the formula structure, constructed one BST (Baseline Structure), achieved structure description, it's an adaptive method for most of the formula type [3]. However, it is restricted to detect structure correctly for formula that has multi-baseline.

In this paper, we propose modified recursive projection profile cutting method to improve character segmentation and the method how multi-baseline could be detected. The proposed scheme could solve problems, mentioned above, and the recognition result could be improved better.
The rest of paper is organized as follows: Section II reviews relevant work. Section III describes character segmentation method using double linked list, modified recursive projection cutting method and classifying method based on neural network. Structure analysis by detecting multi-baseline is described in Section IV. Experiments and results are presented in Section V, and this paper is concluded with a future research plan in Section VI.

2 Related Work

Nowadays, there are many researches on mathematical formula recognition. Among them, the recognition method is divided by BST and projection profile cutting method. BST, an abbreviation of baseline structure analysis is to recognize the character in baseline. Subsequently, each character is classified by position based on baseline. On the other hand, projection profile cutting is to segment two-dimension images to each character by projecting toward $x-axis$ and $y-axis$ alternatively.

2.1 Baseline Structure Tree

The meaning of mathematical formula could be changed by position as well as shape. Thus, each character has to be classified by position. For correct classifying, character region could be designed like Fig. 1. The field of text node is the region of baseline, which has following seven kinds of type: Above, Below,

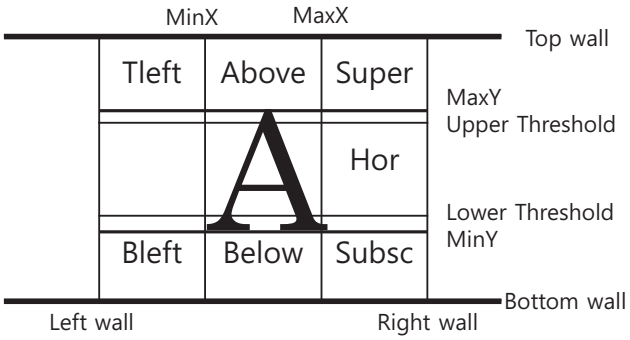

Fig. 1. Character region design.

Super, Subsc, Tleft, Bleft and Hor [3].

Using character region design, formula is represented by base structure trees. Components of base structure tree could be divided by character node and field node. Character node means the character which is located at baseline. And field node means the character which is located side of character region centered by character node. Equation (1) could be represented as BST like Fig. 2.

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{1}$$

In the structure tree, 'x', '=', '-', '$\sum$', 'x' is designated as character located at baseline and they are called as parent nodes. And child nodes are defined

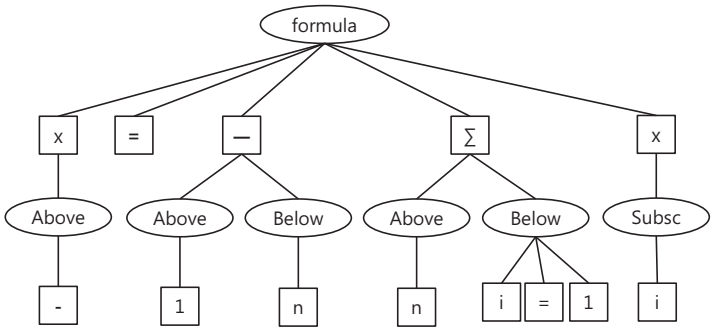

Fig. 2. Structure analysis process.

nodes that have parent nodes. There are a number of advantages to structural representation via this baseline structure tree. The recognizing sequence could be easily found from left to right through baseline detection. Also, using baseline structure tree, the characters that have same depth and same parents could be grouped by binding '(' and ')'. By doing this, transforming to one-dimension texts works more easily. However, the character relation of off-line formula is so uncertain that some characters may not be classified any field. To solve this problem, the field region is overlap with other field. Using this method can avoid recognition failure caused by uncertain position relation [4]. However, although there are multi-baselines in formula, this method assumes that there is only one baseline. To solve this problem, other additional processing is needed.

2.2 Recursive Projection Profile Cutting Method

In formula, the character's sequence for reading is not linear so that recognizing properly is very complicated. By projection profile cutting method, proper sequence of formula could be detected. Histogram could be drawn by projection horizontally or vertically. Next, this method cut the image along with positions whose number of pixels has under threshold value. This method is repeated until there are no positions whose number of pixels have no under threshold value any more.

Fig. 3 represents structural expression by using recursive projection profile cutting method. By this method, segmentation sequence is '∞', '$\sum$', 'n', '=', '0', 'x', 'n', '=', '1', '-', '1', '-' and 'x'. As the form of this formula in LaTeX is $\backslash sum\_\{n = 0\} \wedge \{\backslash infty\} x \wedge n = \backslash frac\{1\}\{1 - x\}$, this segmentation sequence is almost correct. However, this approach is not appropriate to detect superscripts, subscripts, matrices, limit expression (e.g. summations) or expressions within square roots, each of which requires additional processing [5].
Furthermore, they have the chance that two or more character would be recognized as one character based on histogram. Therefore, there are many limitation of recognizing formula by using only recursive projection profile cutting method.

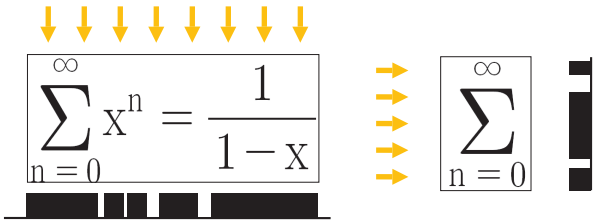

Fig. 3. Character region design Cutting Method horizontally(*left*) and cutting method vertically(*right*).

3 Character Segmentation

3.1 Applied Double Linked List

General character segmentation is implemented according to priority of position, from top to bottom, and left to right. However, when this method is used at formula recognition, they are chances of making error in the arrangement of words. To overcome this problem, we use double linked list to correct sequence between characters easily.
Fig. 4 and Fig. 5 show how double linked list is applied. Fig. 4 represent the structure that memorizes feature of each character. In this structure, position, size and normalized image's pixel value could be saved. And to access adjacent character or to change character sequence easily, address variables like 'prev' and 'next' are defined. Fig. 5 represent the whole structure of formula. It is composed of structure of one character. All variables and functions related to recognition are also declared in this class. (e.g. linked lists' insertion, deletion, addition, binarization, interpolation for normalizing image of each character etc.)

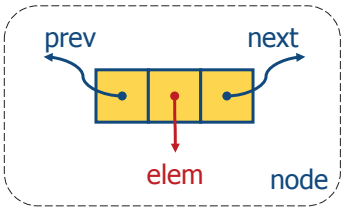

Fig. 4. Structure of one character.

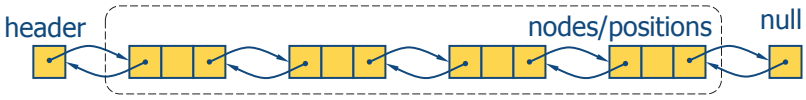

Fig. 5. Whole structure of formula.

3.2 Modified Projection Profile Cutting Method

When previous projection profile cutting method is applied, it could be at risk for recognize two or more characters as one character. This problem is described in Fig. 6. When projection profile cutting method is used, trough between two characters' histogram could be over the threshold value designated so that these characters are recognized as one character.

$$x(t) = \sum_{n=0}^{\infty} x(n)\,\mathrm{sinc}\left(\frac{\omega_s(t - nT_s)}{2\pi}\right)$$

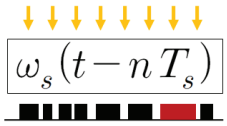

Fig. 6. A projection cutting problem based on histogram.

Although one character is segmented by projection profile cutting, this cannot be always convinced so that additional processing is needed. The solution is labeling after projection. This problem is almost happened during projection not vertically but horizontally. Therefore, labeling sequence has priority from left to right than from top to down. So, it could be overcome by using labeling after projection.

3.3 Classifying individual characters or words

One character could be recognized after using projection profile cutting method followed by labeling. This character could be translated text for representing in LaTeX. The general formula character is shown in table 1[4].

Table 1. The general formula character samples.

Number	Description	Character example
1	Digital number	0123456789
2	English character	ABCDEFGHIJKLMNOPQRSTUVWXYZ abcdefghijklmnopqrstuvwxyz
3	Greece character	ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ αβγδεεζηθθικλμνξοπϖρϱσςτυφϕχψω
4	Mathematical character	+- ×÷/±^()||[ ] =<> % ∗ &# ∈ ∑∏∞∫ ∮ ∀¬ ∨∴∵≦≧≒≠
5	Special character	„.?”:;’∂∇Δ∝

To classify characters, characters that have same histogram distribution must be thought as separate groups. A horizontal axis is divided into three equal parts. Among each part, assign 1 when the number of pixels has more pixels than average number of pixels. Otherwise, assign 0. Therefore, each character could be grouped as 8 groups from “000” to “111”. Also, same process is applied to a vertical axis. The total number of group could be 64. In this process, each character is allowed being overlapped multiple groups. Some examples of this process are described in Fig. 7. ‘4’ is classified “001” vertically and “010”’ horizontally. ‘+’ is classified “010’ vertically and “010’ horizontally.

Artificial neural network is applied to classify and give a name to express text among characters in same groups. As each character is normalized 16 × 12

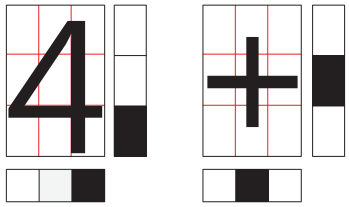

Fig. 7. Example of classifying one character '4' and '+'.

sizes, 16 × 12 input neurons are used. The number of neuron in hidden layer is 150. And, the number of characters in each groups could be the number of output neuron. This structure is described in fig. 8.

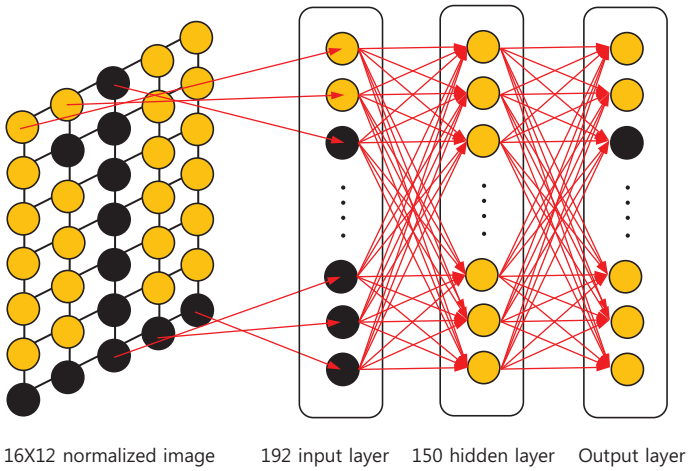

Fig. 8. Neural network structure of character recognition.

4 Structure Analysis

General character recognition is completed by classifying. Though, in case of formula recognition, the position of each character is important as the meaning of position determine whether scripts or not. Therefore, classifying has to be followed by structure analysis.

4.1 Baseline Detection

Baseline Detection method proposed by Zanibbi and Blostein could be used in single baseline [3]. By using this method, superscripts, subscripts and matrices

could be classified in a simple formula. However, there are formulae that have many baseline like (2).

$$\frac{(x-m)^2}{a^2} + \frac{(y-n)^2}{b^2} = 1 \tag{2}$$

To detect multi-baseline, a depth of recursive projection profile cutting function is saved. Depth means that how many function is called until no more projection is excuted. The segmentation order above formula is '(', '-', 'm', ')', '2', '_', 'a' and '2' by projection profile cutting method. The depth of each character is described in Fig. 9.

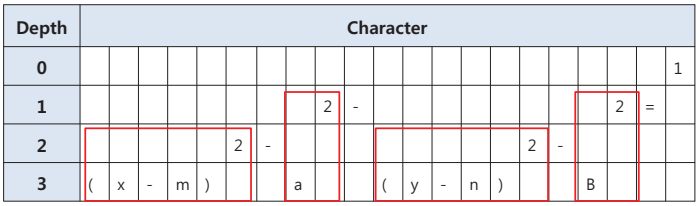

Fig. 9. A formula that has multi-baseline.

Each baseline could be detected from depth information. Baseline is started from depth-increasing point and ended depth-decreasing point. This example is represented in Fig. 9. Each detected baseline could be recognized using BST method. This method could be a solution of single BST method.

4.2 Correcting Relation

The sequence of recognized order using projection cutting method, labeling and BST method couldn't be always reasonable. Some structurers which contain Σ, $\prod$, etc. in formula have to be modified. Generally, the sequence of equation is symbol, superscript and subscript. However, the sequence of recognized order for equations included Σ or $\prod$ is superscript to symbol followed by downscript. These special characters have to be executed by exception processing. The relationship between characters is represented by double linked list. Therefore, change of order is easily executed by changing address which indicates about previous or next character.

5 Experiment result

We carried out experimental recognition for a wide variety of off-line mathematical expressions. Fig. 10 shows the original images used and the recognition results reproduced by LaTeX.

Most formula could be recognized correctly. However, some formula failed to recognize partially. Last formula in Fig. 10 failed to recognize ωt and kz. These fail came from narrow spaces between two letter. It means that two separate letter was recognized one letter during projection.

Original Image	Transformed Text Format	Representation in LATEX
$emf = \oint EdL = -\frac{d}{dt}\int BdS$	emf=\oint EdL=- \frac{d}{dt}\int BdS	$emf = \oint EdL = -\frac{d}{dt}\int BdS$
$x(t) = \sum_{n=0}^{\infty} x(n) sinc(\frac{w_s(t-nT_s)}{2\pi})$	x(t)=\sum_{n=0}^{\infty} x(n)sinc(\frac{\omega_s (t-nT_s)}{2\pi})	$x(t) = \sum_{n=0}^{\infty} x(n) sinc(\frac{\omega_s(t-nT_s)}{2\pi})$
$\nabla \times H = J + \frac{\partial D}{\partial t}$	\bigtriangledown \times H = J + \frac{\partial D}{\partial t}	$\nabla \times H = J + \frac{\partial D}{\partial t}$
$\frac{(x-m)^2}{a^2} + \frac{(y-n)^2}{b^2} = 1$	\frac{(x-m)^2}{a^2} + \frac{(y-n)^2}{b^2}=1	$\frac{(x-m)^2}{a^2} + \frac{(y-n)^2}{b^2} = 1$
$c_n = \int_{T_0} x(t) e^{-jnw_0 t} dt$	c_n=\int_{T_0} x(t) e^{-jn\omega_0 t} dt	$c_n = \int_{T_0} x(t) e^{-jn\omega_0 t} dt$
$x[n] = x(t) \sum \delta(t - n T_s)$	x[n]=x(t)\sum \delta(t-nT_s)	$x[n] = x(t) \sum \delta(t - nT_s)$
$1 + x + x^2 + x^3 + \ldots = \sum \frac{1}{1-x}$	1+x+x^2+x^3+\cdots = \sum \frac{1}{1-x}	$1 + x + x^2 + x^3 + \cdots = \sum \frac{1}{1-x}$
$\varepsilon_x(z;t) = \mathrm{Re}[E_x^- e^{j(\omega t + kz)}] = E_{x0}\cos(\omega t + kz + \phi_x)$	\epsilon_x(2;t)=Re[E_x ^{-} e^{j(a+b)}] = E_{x0}cos(a+kz+\phi_x)	$\epsilon_x(2;t) = Re[E_x^- e^{j(a+b)}] = E_{x0}cos(a + kz + \phi_x)$

Fig. 10. Experiment results.

6 Conclusions

This paper proposed a modified recursive projection cutting method with labeling and demonstrated its effectiveness through the experiment with neural network. To classify normal letter, superscript and subscript, depth of recursive projection was used. Altough most of formula could be recognized correctly, there are some problems to solve. As the space between two characters is narrower, it would be harder to recognize correctly so that projection method is very sensitive to noise. Therefore, it is also important to reduce noise in preprocessing to yield results of high accuracy for recognition. Another imporant problem is that there are many similar characters in formula expression, for example, 'a' and 'α'. These kind of letters couldn't be always recognized without knowing context of formula. Therefore, it is needed to comprehend contextual

information by analyzing adjecent letters.

In future research, these two problems will be the main issues in formula recognition. As well as off-line recognition, this research could also be extended to on-line formula recognition.

Acknowledgements

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Human Resources Development Program for Convergence Robot Specialists support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-H1502-12-1002)

References

1. Lin XY, Gao LC, Tang Z, Lin XF and Hu X (2011) Mathematicalformula identification in PDF documents. Paper presented at Int Conf Document Anal and Recognition, Beijing, China, pp 1419–1423, Sep 2011. doi: 10.1109/ICDAR.2011.285
2. Okamoto N and Miao B (1991) Recognition of mathematical expressions by using the layout structures of symbols. Paper presented at Int Conf Document Anal and Recognition(1), Saint-Malo, France, pp 242–250, Sep 1991.
3. Zanibbi R, Blostein D and Cordy J (2001) Baseline structure analysis of handwritten mathematics notation. Paper presented at Int Conf Document Anal and Recognition, Seattle, WA, USA, pp 768–773, Feb 2001. doi: 10.1109/ICDAR.2001.953892
4. Li Y, Wang K, ShangGuan W and Tang L (2008) The research of mathematical formula recognition method base on baseline structure analysis. Paper presented at Int Conf Internet computing in Sci and Eng, Harbin, China, pp 53–59, Jan 2008. doi: 10.1109/ICICSE.2008.102
5. Zanibbi R (2000) Recognition of mathematics notation via computer using baseline structure. Technical Report ISBN-0836-0227-2000-439, Dept. Computing and Information Science, Queen's University, Kingston, Ontario, Aug 2000.