Journal of Theoretical and Applied Information Technology
31st January 2013. Vol. 47 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN ANALYTIC HIERARCHY PROCESS BASED METHOD TO PROCESS MATHEMATICAL EXPRESSIONS

[1]LEI GAO, [2]SHULIN PAN, [1]SHEN JIAO

[1] School of Computer Science, Southwest Petroleum University, Chengdu 610500, Sichuan, China

[2] School of Resources and Environment Engineering, Southwest Petroleum University, Chengdu 610500, Sichuan, China

ABSTRACT

Mathematical expressions exist in all kinds of documents. The automatic location , recognition, analysis and understanding of mathematical expressions are important. Recognition of mathematical expressions has been an attractive problem for the pattern recognition community. When we meet the mathematical expressions we can't recognize them like the way of ordinary text recognitions. We are in urgent need of a new method to process the mathematical expressions. We should firstly recognize all the symbols which constitute the expressions and define the composite structures and constrains and relationships of all the symbols. And we also should distinguish the levels of hierarchy of every symbol and the corresponding matrices and select the feature items and set the weight numbers. Analytic hierarchy process is caught to build the expression model. First, the hierarchy model is set up. Second, pairwise comparison matrix is set up. Then, a series of computer process will be going on including computing the weight vector and consistency checking and computing combination weight vector and doing combination consistency checking and constructing the judgment matrix and computing weight vectors. Experience has proved that this is a proper method to process mathematics expressions.

Keywords: *Mathematical Expressions, Composite Structures, Levels of Hierarchy, Weight Numbers*

1. INTRODUCTION

Mathematical expressions exist in all kinds of documents. Thus, the automatic location , recognition, analysis and understanding of mathematical expressions are important [5].

However, because of the many differences between mathematical expressions and normal texts, expression processing is far more difficult than ordinary text recognition [5]. Recognition of mathematical expressions has been an attractive problem for the pattern recognition community[1].

There is a wealth of mathematical knowledge that could be potentially very useful in many computational applications[8]. However, the attempts are attended by the momentous results only in the image processing as follows. The document [10] deals with segmentation and recognition of touching characters appearing in scanned mathematical expressions. The technique is based on multifactorial analysis that integrates several factors determining cut-positions in a touching character image. The document [1] suggests a method to deal with the problems such as enormous uncertainties and ambiguities as encountered during parsing of the two-dimensional structure of expressions. [5]This paper reviews the history of mathematical expression image processing, presents the modules of expression processing, surveys the primary processing methods of expression extraction, expression recognition, expression analysis and performance evaluation, and gives the perspective of future research directions.

It was clear at first glance that at present most of methods of mathematical expressions processing are based on image processing. This means that when we meet the mathematical expressions we can't recognize them like the way of ordinary text recognitions and we should think of the mathematical expressions as images.

We could easily find out the trouble that if we process the mathematical expressions as images, recognition rate is an inevitable exist. So many expressions won't be recognized correctly. This can't clearly meet the development of computer. We are in urgent need of a new method to process the mathematical expressions.

The new method establishes the model of the structure of mathematical expressions and

Journal of Theoretical and Applied Information Technology
31st January 2013. Vol. 47 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

distinguishes the levels of hierarchy of every symbol and the corresponding matrices and selects the feature items and sets the weight numbers .So the model of the spatial relationship among all the symbols on these grounds can be established. At last, we bring forward the method of processing the mathematical expressions based on the structures of symbols and the models.

2. THE RULES OF EXPRESSIONS DEFINED

To process the mathematical expressions correctly, we should firstly get all special symbols into shape .

Before we establish the model of mathematical expressions , we should firstly recognize all the symbols which constitute the expressions. The symbols that the keyboard has can easily been judged, while special symbols including complex symbols are hard to be actually discerned and distinguished. Each subject has its own special symbols. Table 1 lists some special symbols of Mathematics and Physics.

Table I: Special Symbols

Subjects	Special symbols
Mathematics	≌\∪\∮
Physics	Ω\ρ\△

So we should establish the library of all special symbols which we may come across.

An expression is expressed by vectors designing the feature weights as the components. All the contents should be designed including the contents ,fonts and characters of all the symbols and the relations among the symbols. These symbols include special symbol cluster and normal symbol cluster. Special symbol cluster means the symbols which can't be clicked by the keyboard and normal symbols means the symbols which can be clicked by the keyboard.

An expression consists of operators and operands. We should plan the rules of grammar and call the corresponding rules of grammar based on symbols contents. We should settle the combination relations between symbols and check legalities of symbols.

The priority levels between all operators and the types of operators should also be defined. The number of sub-expressions owed by operators is defined and the composite structures and constrains of all sub-expressions are defined that are used to analyze the priority orders between different operators and smooth ambiguity.

3. ESTABLISHMENT OF THE MODEL

Analytic hierarchy process is caught to build the model. A detailed description of each of these steps follows.

1) First, the hierarchy model is set up. According to different properties we break every related factor into several levels from above to blow. Every factor of same level is subservient to the factors of higher level and simultaneously dominates the factors of lower level.
2) Pairwise comparison matrix is set up. Beginning with the second level pairwise comparison matrix is set up by using Paired comparison method and the measure of comparison from 1 to 9 for every factor of the same level subservient to the factors of higher level.
3) To compute the weight vector and consistency checking. Computing the biggest eigenvalue and the corresponding feature vector of every pairwise comparison matrix and consistency checking by using consistency target, random consistency target and consistency ratio. If the inspection has been got past, the feature vector is the weight vector, otherwise we should reconstruct the pairwise comparison matrix.
4) To compute combination weight vector and do combination consistency checking. If the inspection has been got past, we can make a decision according to the results showed by combination weight vector ,otherwise we should reconsider the model or reconstruct the pairwise comparison matrixes which have large consistency ratios.
5) To construct the judgment matrix. One important characteristic of analytic hierarchy process is to use the ratio of two importance degrees to represent the importance degrees of two corresponding plans.
6) To compute weight vectors. To dip into useful information from the judgment matrix and understand the regulation of objects and provide the scientific bases for a decision, computing weight vectors of the judgment matrixes. The calculation methods to compute the weights of every symbol is to use the formula as follows:

Journal of Theoretical and Applied Information Technology
31st January 2013. Vol. 47 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$W(t,\bar{d})=\frac{tf(t,\bar{d}).\log(\frac{n}{n_t}+0.01)}{\sqrt{\sum\left[tf(t,\bar{d}).\log(\frac{n}{n_t}+0.01)\right]^2}} \quad (1)$$

$W(t,\bar{d})$ is the weight of symbol t in the level $\bar{d}$. $f(t,\bar{d})$ is the frequency of symbol t in the level $\bar{d}$. N is the number of levels. n_t is the number of t in all levels. The denominator is the normalized factor.

After the above steps, the formula can be represented as the m*n symbols-formula matrix:

$$A_{m*n}=[\alpha_{ij}] \quad (2)$$

m is the number of all different symbols including special symbols of a formula. a_{ij} is the weight value

4. THE SYSTEM DESIGN

This system is designed in three tiers.

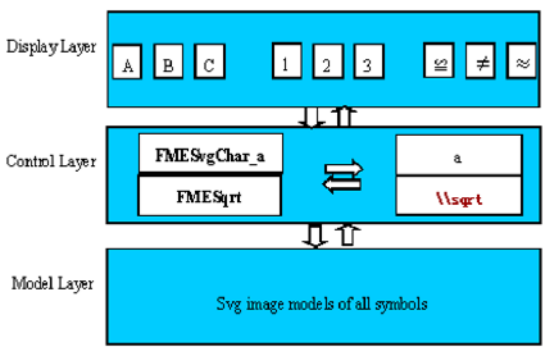

Figure 1: The System Architecture

All the system mainly has four files. The file named button_icon saves the images of buttons. The file named FFEUnits which saves the classes which inherit from FFEUnit. The file named icon_button_skin saves the skins of buttons. The file named svgs saves the vector graphics designed for all characters.

The system involves four basic classes: FFEUnit , FFEScreen , FFEContainer and FFECursor . All the subclasses inherited from these classes constitute the main body of the system , such as containers and formula modules.

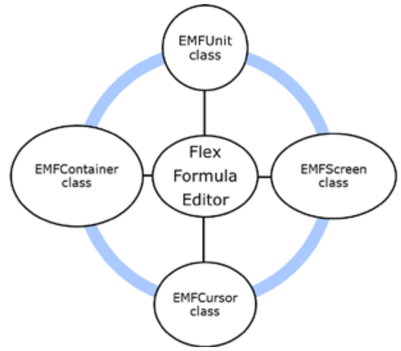

Figure 2: The Relations Of Classes

An expression normally includes many units such as individual characters or a function. That is as follow as equation (1):

$$f(x)=a_0+\sum_{n=1}^{\infty}(a_n) \quad (3)$$

This equation includes two units: individual character a_0 and a function $\sum_{n=1}^{\infty}a_n$.

To recognize every part of each expression ,we put all the components of expressions into some containers and therefore we can recognize all the expressions via the unified unit. Every expression can be separated into its components. This way facilitates the recognition of expressions. Thus , every expression has the same layout via containers. This process is depicted in Figure 3.

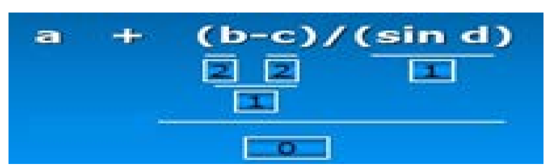

Figure 3: The Containers Of The Expression

In Figure 3, we can easily judge that the expression has two parts: individual character a and two numbers divided . And we can continue to separate these parts. The second part can be divided into the dividend and the divisor. The dividend can be divided into the minuend (b) and the subtrahend (c). And the divisor is a sine function .

Through these procedures this expression become inseparable. For one thing the expression has two parts which are defined as zero order containers. Second again, the zero order container includes two parts and this means (b-c) and (sin d) are the 1 level containers. It is exactly same analogy that b and c are the 2 level containers. Thus the final level containers can be indivisible.

Journal of Theoretical and Applied Information Technology
31st January 2013. Vol. 47 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

So each expression can be divided into inseparable parts and these parts have united units.

Every level container is defined as one instance object of class FFEContainer . And the component modules of each container are defined as FFEUnit.

The heritance relationship and the subclasses of the class named FFEUnit are listed in Figure 4.

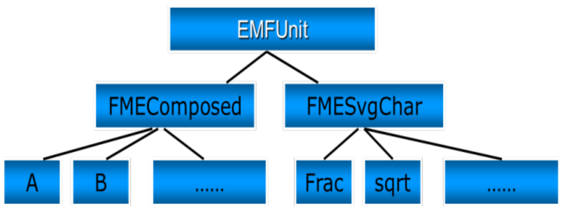

Figure 4: The Heritance Relationship And The Subclasses Of The Emfunit Class

All the symbols of expressions inherit from FFEUnit class which sets the coordinate and size property of every character .

Individual characters inherit from FFESvgChar class. Combing characters inherit from FFEComposed class and consist of several individual units. An individual unit may be an individual character or a combing character.

The FFEComposed class is equivalent to a container which has points of resemblance to the FFEContainer class. Both of them have the function of update which is used to update the content of containers .

The FFECursor class is used to save the information of the cursor such as the start position , the end position and the judgments if this is the selected square. We can make use of this class to identity the location where we should compile expressions by a mouse. The selection of cursors is measured in containers or component modules units.

FFEScreen class defines a series of operations of screens . For instance, after selecting contents in the screen,

5. THE SYSTEM DESIGN

The system is developed based on Flex framework of Flash. The development platform is Flash Builder 4.5 and the development language is Actionscript . All the platform which support Flash player can run this system.

This system is divided into two client ports. The deployments of both the ports is as follows:

A. the teacher client port:

Because of the webpage method of the teacher client port , the system is deployed in the client port as a plugin.

B. the student client port:

The system is also integrated in Flex student client port. The student client port based on Flex framework and the editor will enjoy seamless connection.

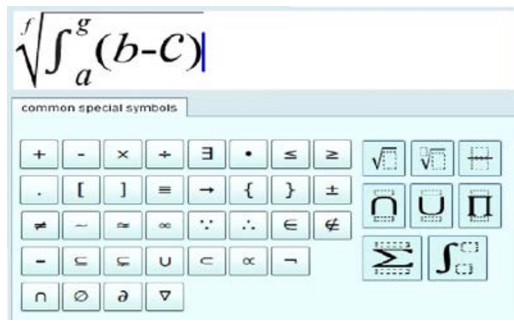

Figure 5: The Interface Of The Editor

6. CONCLUSION

This paper reviews the history of mathematical expression image processing, presents the modules of expression processing, surveys the primary processing methods of expression extraction, expression recognition, expression analysis and performance evaluation, and gives the perspective of future research directions. Experience has proved that this is a proper method to process mathematics expressions.

ACKNOWLEDGEMENTS

This work was supported by the key project of department of education of Sichuan (12ZA196) and the project of department of education of Sichuan (12ZB158) and NSFC (41204101) and the key project of open experiments of Southwest Petroleum University (KSZ1136).

REFRENCES:

[1] Harold Mouchère, Christian Viard-Gaudin, Dae Hwan Kim, Jin Hyung Kim, Utpal Garain, “Competition on Recognition of Online Handwritten Mathematical Expressions”, *Proceedings of 2011 International Conference on Document Analysis and Recognition*, September 18-21, 2011, pp. 1497-1500.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] Jin Wu , Wu-Mo Pan, Jian-Ming Jin, Qing-Ren Wang, "Performance evaluation and benchmarking on document layout analysis algorithm", *Proceedings of the 2nd International Conference on Machine Learning and Cybernetics*, Nov. 2-5, 2003, pp.2246-2250

[3] Chan K.F., Yeung D.Y., "Error detection, error correction and performance evaluation in on-line mathematical expression recognition", *Pattern Recognition*, Vol.34, No.8, 2001, pp.1671-1684

[4] Costagliola G. Chang Shi-Kuo, "Parsing Linear Pictorial Languages by Syntax-directed Scanning, *Languages of Design Journal*, Vol. 2, No. 3, 1994, pp.223-242

[5] Jin Jian-Ming, Jiang Hong-Ying, Wang Qing-Ren, "Survey of mathematical expression image processing", *Pattern Recognition and Artificial Intelligence*, Vol. 18, No. 4, 2005, pp. 429- 440

[6] Lee Hsi Jian, Lee Min-Chou, "Understanding mathematical expressions using procedure -oriented transformation", *Pattern Recognition*, Vol. 27, No. 3, 1994, pp. 447- 457

[7] Lee Hs-Jian, Wang Jiumn-Shine, "Design of a mathematical expression recognition system", *Pattern Recognition Letters*, Vol. 18, No. 8, 1997, pp.289-298

[8] Benjamin P. Berman, Richard J. Fateman, "Optical Character Recognition and Parsing of Typeset Mathematics", *Journal of Visual Communication and Image Representation*, Vol.7, No. 1,1996, pp.2-15.

[9] Jin Jian-Ming, Jiang Hone-Ying, Wang Qing-Ren, "Survey of mathematical Expression image processing", *Pattern Recognition and Artificial Intelligence*, Vol.18, No.4, 2005, pp.429-440.

[10] Garain U tpal, Chaudhu ri B. B, "An approach based on multifactorial analysis", *Proceedings of the 8th International Conference on Document Analysis and Recognition*, August 29 - September 1, 2005, pp.177- 181.

[11] Jin Jian-Ming, Jiang Hong-Ying, Wang Qing-Ren, "Mathematical Expression Recognition System : MatheReader", *Chinese Journal of Computers*, Vol. 29, No.11, 2006, pp.2018- 2026