

# The Study of Mathematical Expression Recognition and the Embedded System Design

Qi Xiangwei

Xinjiang Normal University, Urumqi, China

Email: xiangweiqi10@163.com

Yusup Abaydulla

Xinjiang Normal University, Urumqi, China

Email: 47266861@qq.com

**Abstract**—Mathematical expression recognition is an important problem about pattern recognition, because mathematical expression is an essential part of scientific literature. This paper focused on two aspects, first of all, About the recognition of on-line handwritten mathematical expressions, this paper elaborates the structural analysis strategy from the perspective of the general mathematical expression, we bring forward a feasible solution to the overall structural analysis by using some pretreatment methods, such as dominating relationships, hard-regulations, function merging, and so on. The results show that this method can overcome some shortcomings of traditional structure analysis methods, and also it can dispose some recognition errors that often appear in the space structure and logical structure of the on-line handwriting recognition system effectively, such as symbol overlapping and handshaking, complex layer structure, the table structure and so on. Secondly, on this basis, We have contrived an online handwriting recognizing system of mathematical expression based on an embedded equipment, which supports in the use of the languages of the minority, such as Uyghur and Kazakh languages .For the features of mathematical expression recognizing, and from the point of ensuring the system's efficiency , this paper has constructed an embedded platform, on which the capability of the system has been tested, and it has compared the data gained from the test with those in PC , which offers a gist to improve the algorithm and perfect the system.

**Index Terms**—general mathematical expression, handwriting recognition, analysis of the structure, the embedded system, solutions

## I. INTRODUCTION

Mathematical expression recognition is an important problem about pattern recognition, because mathematical expression is an essential part of scientific literature. In off-line recognition, handwritten or printed expressions are given in the form of images or bit-maps, a static representation of the data. In on-line recognition,

computers with pen devices (graphic tablets, contact sensitive whiteboards, Tablet PCs) store and record the data by the way of digital ink and a dynamic representation.

This paper analyzes the structure of the general mathematical expression, because after the character recognition device completes pre-processing of data, segmentation and character recognition, the next task is to analyze these characters from the spatial structure and logic structure to determine their spatial and logical relationship , and then form a no ambiguity mathematical expression at last[1].

Mathematical expression is a complex two-dimensional structure, if all the characters are divided appropriately, then we will be able to express mathematical expression structure by hierarchy process, whereas even all the characters are correctly recognized, appropriate division is also very difficult. Most of the former work in structure analysis is done in the subset of mathematical expression, that is, these methods are used to do structural analysis under certain conditions, for example, the method of DCG (Definite Clause Grammar) [2], is totally based on algebraic expressions. When the operator in the mathematical expression is arrayed to a line, we can use priority of operators to determine the group of characters, if the operators are the non-linear arranged, the dominating field of operators determine the

group of characters. For example, in  $x + \frac{c}{b}$ , the expression is interpreted as  $x+(c/b)$ , so the operator "+"

determined"-"; however, the expression  $\frac{a+b}{c}$  was interpreted as  $(a + b) / c$ , because the operator "—" dominate "+",because the latter is in the former's dominating region. Obviously, if the operators in a mathematical expression can not be understood by computer, are just some symbols of a certain structure or an object, for example  $a^+ \rightarrow b^+$ , these methods and interpretations would be meaningless, and this is also Chen advanced when the range of recognize expression has changed, the effect of the methods will be cut down and even can't be used [3]. Therefore, it is necessary to

This study is funded by National Natural Science Foundation Project (No.60463005) and the Xinjiang Normal University graduate students scientific and technological innovation fund (No.20081201). Corresponding author: yusup, E-mail: 47266861@qq.com.

create a general mathematical expression structural analysis methods, Zanibbi[4] brings forward the expression and analysis methods of Baseline Structure Tree, he provides a feasible approach to analyze the structure of the general mathematical formula; but he does not consider the analysis of the table structure, at the same time this method is of quite low effecton in the whole expression structure analysis, only 39% correct rate. Tapia[5] improved this method by using minimum spanning tree and regional dominance, but calculating computing functions of two characters' nodes is complex, which is difficult to adapt to variable expressions, simultaneity this method does not consider the expression's table structure, and neither have an effective method of structure logical analysis. For example, when one main character has upper and right-upper relations, which are all considered as upper relations, regardless of whether they form a composite or not. This paper will analysis and discuss structure analysis strategies from the perspective of general mathematical expression.

II. DOMINANCE RELATIONSHIP

The range of symbol is the desired location of operation, Chang[6] brought forward Symbol Dominance and the Dominance Range, etc. used in structure analysis of mathematical expressions, that is, if a character "s" dominates a character "a", then "a" is in dominance range of "s", and "s" isn't in dominance range of "a". Regard to shortcomings of judging the spatial relationship which depend on centroid and range, this paper utilizes Chang's method of defining domination and the form of relationship to define the character domination.

Definition 1: Dominance relation

The relation and operating advantages of mathematical symbols are precise or ambiguous determined by symbol location and relative size in expressions. Spatial location of the top-left, above, superscript, right, subscript, below, below-left and sub-expression are used to determine these relationships. For example, fraction expression contains numerator, the denominator and the horizontal is considered standing on the horizontal line above and below.

As to one or more spatial relationships that are appointed by character "s", if "s", as a main character and a, as an object, form one of these spatial relationships, then we said s and a form dominating relationship. See figure 1:

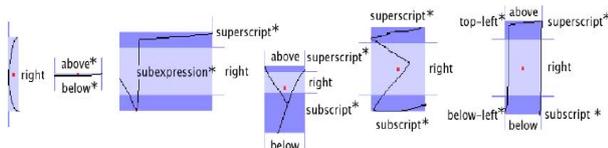


Figure 1. Spatial relationship of characters

If s, as the main character, and a form a dominant relationship, while a, as the object character, and s do not form dominating relationship, then s dominate a.

Definition 2: non-dominating relationship

Via this definition we connect the dominance relationship and the space relationship among characters,

the reason why do so is to give up the methods of regional demarcation and centroid when judge the spatial relationship between characters and symbols[7]. Table I lists some of the dominant relationship of characters:

symbol	dominance relation
Binary operator, the brackets and ';;', etc.	right level relation
Closed brackets	right-up relation
radical sign	right-up, left-up
numbers	right-up
fraction operator	up, down
Variable character	left-up, up, right-up, left-down, down, right-down
some functions, such as 'lim', etc.	right-up, left-down, down, right-down
⌋ ⌋ ⌋, etc.	non-dominance
The remaining characters	except left level, right level

III. THE FIXED-REGULATIONS OF MATHEMATICAL EXPRESSIONS

Mathematical expression is a kind of natural visual language which conveys the mathematical meaning by using space relationship between characters and symbols[8]. Therefore, there isn't normal definition, however, because of long-term use of mathematical expressions, people have created some inherent rules. For example, the pioneer of Mathematical expression recognition, Anderson[9] points out the significance of the direction of the mathematical expression to identify the structure, that is, if we deal with it along with the baseline of expression from left to right, we can get a grammar structure of an ideal linear expression, for example, x+y+z/2. Therefore, the direction from left to right we read the expression is a fixed regulation.

Another fixed regulation exist in the no ambiguity mathematical expression, the leftmost characters in the mathematical expression are the beginning characters. However, there are exceptions, we summed up the following several cases in this study [10, 21]:

(1) Fraction line in few cases may not be at the leftmost. As shown in Figure 2-a:

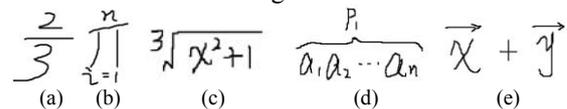


Figure 2. The special start symbol

(2) Some variable characters, such as "∑ ∏" etc., may not be at the leftmost of the expression, but often the beginning characters of the expression. As shown in

figure 2-b, "i" is on the left of "∏", but we can not make "i" as the beginning characters of the expression.

(3) The top-left relation of radical sign can not be used in the beginning of the expression, As Figure 2-c:

(4) Some non-script characters, such as some brackets which indicate super or sub relationship, such as "⌋ ⌋ ⌋", and some symbols which express direction of the vector such as "→←" and so on, these may be

the leftmost characters in the expression, but it can not become the beginning characters of the expression, as Figure 2-d, e.

Chang[6] brought forward that if a character *s* dominates a character *a*, while *s* is not at the dominate range of *a*, in this case, the dominant character should be the beginning characters of the expression, shown in Figure 2-a fraction line dominates "3", so fraction line can be regarded as the beginning characters of the expression. By far, this approach is the main approach to position beginning characters in some expression recognition system. However, Martin[11] listed an ambiguous case, that is, when *s* dominates *a*, and also *a* dominates *s*, we can not determine which characters should be the beginning characters in the expression, as shown the example in Figure 3. In figure 3-a, the above fraction line dominant the below fraction line, and vice versa, and at this time we can not determine which fraction line is the beginning character in this expression. In Figure 3-b, "e" is in the dominating range of the integral symbol, while integral symbols are also in the dominating range of "e". concerning these ambiguity questions, Tapia and others[5] make a distinction by the characteristics between characters and symbols, that is, making fraction line of a higher center of mass as the dominant character in the expression, for other types using the larger font size character as a dominant character. This approach can also solve some dominating ambiguities which are often seen

in mathematical expression, for example,  $\sum_{x=1}^{100} x + 1$ , according to the size of the characters, we can identified " $\sum$ " as the beginning character of this expression.

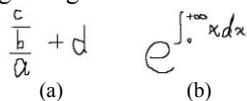


Figure 3. Ambiguity examples of the beginning characters

By applying the dominating relation and ambiguity decision rules, this paper determine the beginning characters of mathematical expression by the function FindStartSymbol().

IV. FUNCTION COMBINATION

We have mentioned above in this paper we study the relationship of space between characters and symbols is aimed at space relationship between characters, because there are many functions in the mathematical expression, it is meaningful that only when these functions are considered as one character, otherwise they would affect the later logical structure analysis, because in the process of structure analysis, just considering these individual characters in function would make space structure very

messy. For example,  $\lim_{x \rightarrow \infty}$ , if we just use a single character to denote structure, then the lim which denoting the function limit and the lower part  $x \rightarrow \infty$  will be identified as many types of structures, so as to miscarriage of justice to the structure of the

expression[12,21]. Therefore, these functions in mathematical expression should be regarded as a whole, this paper use FindFunction () to accomplish this function. We have summed up some common functions, as shown in table II:

TABLE II. FINDFUNCTION () CAN FIND SOME FUNCTIONS

lim	log	exp	sin	cos	tan
prob	avg	in	arcsin	arccos	arctan
cot	sec	csc	max	min	
arccot	arccot	lt	arg	ln	

V. SPATIAL STRUCTURE ANALYSIS OF EXPRESSIONS

A very difficult problem in structure analysis of handwritten mathematical expression is to deal with irregular layer structure. These irregular structures may lead to serious errors in expression structural analysis. Just as most ways, this study suggests that character recognition has been completed correctly before structural analysis. Because the application of the attributes of characters and symbols can obviously increase the effect of the expression structure analysis, of course, one character recognition device can not meet such a request, it usually need to add the human-computer interaction function to dispose error characters in the stage of character recognition ,in order to correct the mistaken recognized characters[13,16] .

We divided the structural analysis of mathematical expression into the following 5 major steps:

(1) Find functions from mathematical expression by FindFunction(), assemble them into a whole, and regard it as a character. We take the function after disposal as one character.

(2) To extract the characters in radical sign from expression, and constitute the inner character mark for each radical sign. These characters as a list are added to expression list Expression, they do not participate the overall structure analysis of the expression this time.

(3) To extract complex. The extract method of present complex is simple, which is mainly characterized by the main subject and size of complex. If a character can be as a character of the main body of the complex, and its font size is twice larger than its left character or its right character, seeing that it is a complex, and can find another character pair in the relative same size, taking the middle part as a part of the complex; If there is no another corresponding characters pair, when it is an open bracket, extracting the final expression, if it is a close bracket, extracting the beginning part of the expression. Using the method of complex segmentation to divide them, and then take each element as an expression to add to the Expression.

(4)To complete the expression structure analysis in expression.

(5)To constitute a complete expression.

We call that the expression disposed step (1) (2)(3) pre-post expression; the following structural analysis does not include the disposal of internal relations and the complex internal structure, because the inner relationship

in radical sign and complex expression elements can be taken as one expression.

VI. STRUCTURE ANALYSIS OF THE EXPRESSION AFTER THE PRE-DISPOSE

Handwritten character sequence in mathematical expression constitutes an expression through a two-dimensional array, when we get a catena of a character, we can obtain its identity, the border attribute and all the coordinate data of this character, these are basis to do expression structural analysis. First of all, we will briefly introduce some definition of fractal in structure analysis of mathematical expression [14, 17]:

(1)The first type of characters: binomial symbols, fraction symbols, braces, radical sign, matrix symbols, range symbols in the condition expression.

(2)The second type of characters: other symbols except the first type of characters.

(3)Simplex fractal: the adjacent second type of characters or character strings at the same level; it does not include relation characters except horizontal relation. Such as " $x^2 + a$ ", " $x$ " is a simplex fractal, " $+ a$ " is a simplex fractal, " $2$ " is also a simplex fractal.

(4)Main body: A second type of characters or the adjacent first type characters or strings locate in the same level, these characters or the middle character of strings contain the characters of one or more relationship in addition to the level relationship. Such as in " $x^2 + a$ ", " $x$ " is a main body; in " $\sqrt[3]{y + x^2}$ ", " $\sqrt$ " is a main body, " $x$ " is also a main body.

(5)Basic fractal: in the mathematical expression, a simplex fractal or a first type character referred to basic fractal. in " $y = \sqrt{x} + 1$ ", " $y =$ ", " $+ 1$ ", " $\sqrt$ ", " $x$ " are basic fractal.

(6)Complex fractal: two or more Basic fractal form a complex fractal, a complex fractal can be disassembled to basic fractal.

(7)Object: forming relation fractal with the main body, the fractal can also be a character or a symbol.

Using fractal can express a mathematic expression to a one-dimensional text structure, taking the main body as the basic, by the order from left to right, complex to simple, from top to bottom, to conduct fractal [15, 21]. Figure 4 is a simple example of fractal; Figure 5 is the further fractal and fractal number. Comprehensive decomposing of mathematical expression can be done by this kind of fractal.

$$f(x) = \frac{\begin{matrix} 2 & 3 \\ 5 & 4 \end{matrix}}{\sqrt{x^2}} + \overline{AB} + \int \sin(x)$$

Figure 4. Fractal

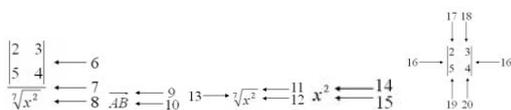


Figure 5. Further fractal

A mathematical expression is connected from left to right by a variety of fractal, the first main body of these fractal connect each others from left to right in horizontal direction, forming a main body structure, so structure analysis must confirm the first main body of these fractals, in order to fix the main structure of the expression, and then determine the characters included in every fractals, and to identify the spatial relationship between the main body and the other characters. As the region of fractal and the space relationship are not one-to-one correspondence,

for example, in " $\int_0^{1002} x dx$ ", " $1$ " is the upper-left relationship of " $\int$ ", " $2$ " is the upper-right relationship of " $\int$ ", but " $1$ " and " $00$ " and " $2$ " should be see as a whole, upper relationship of " $\int$ ". So in the analysis of fractal, we must consider how to incorporate the characters with different spatial relationship in order to construct the corresponding fractal and determine the logical structure. Only when we have determined the character series of every object of the fractal, we can do the further expression structure analysis to object.

In addition, it often appear such table structure in the mathematical expression, for example, in " $\sum_{\substack{x \in (0, \infty) \\ y \in (0, 1)}} (y/x)$ ", " $\sum$ " and " $x \in (0, \infty)$ " and " $y \in (0, 1)$ " form down fractal, " $x \in (0, \infty)$ " and " $y \in (0, 1)$ " are a kind of the table structure, this table structure must be divided, otherwise, it would cause serious error in structural analysis.

Therefore, the process of expression structure analysis is divided into 4 steps after pretreatment:

The first step is to find the beginning character of expression.

The second step is to find the first main characters in all fractals of expressions.

The third step is to analyze the logical structure of the expression, that is, to determine characters and the grouping of characters in every fractal.

The fourth step is to identify whether there are the table structures in every object of fractals or not, if so, then we separated the table structure into different expressions.

The following are details:

A. Identify the start characters

Identifying the start characters is the base of structure analysis of a mathematical expression. According to fixed-regulations mentioned above, a mathematical expression should follow the left-to-right rule in reading and writing. Normally, the most left character is the starting character, according to what we have described above, dominating relationship should be used to identify the most-left starting character, meanwhile, we use the size of character and the height of centroid to solve the problems of ambiguity [12,18].

First, collating the order according to the least x coordinate of the character in the expression, we can gain a group of characters, calling it MathSmybol= $C_1, C_2, \dots, C_n$ , if  $i < t, \min x(C_i) < \min x(C_t)$ . Through the function FindStartSymbol() in which the parameter denotes a list

files of characters , we can find the initial character of the expression. Algorithm is described as follows:

FindStartSymbol(MathSmybol):

1.  $n \leftarrow \text{Len}(\text{MathSmybol})$  ; Get the number of characters in the list.
2. If  $n=1$ , return
3. If  $S_n$  and  $S_{n-1}$  is the dominating relationship, delete  $S_{n-1}$  from the list and return 1;
4. If  $S_{n-1}$  and  $S_n$  is the dominating relationship, deleting  $S_n$  from the list and return 1;
5. The relationship between  $S_n$  and  $S_{n-1}$ , and between  $S_{n-1}$  and  $S_n$ , neither is dominating, delete  $S_{n-1}$  from the list and return 1;
6. If the relationships between the two are mutual dominating, according to the rule of ambiguity, delete from the list;
7. Return to the first step;

Figure 6 is the starting characters found by FindStartSymbol(), expressed by dashed frame:

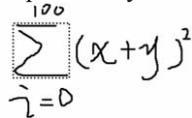


Figure 6. The starting character found by FindStartSymbol()

**B. Identify the main characters of all fractals in the expression**

The first layer of mathematical expression connect from the left to right constituting the main baseline[4,19,20,21], regard the list of characters as MBS, we can obtain these character list with function GetMBS(MBS,MathSymbol) . Algorithm is described as follows:

GetMBS (MBS): MathSymbol,

1. If MBS is empty, use FindStartSymbol (MathSmybol) to find the initial characters, and join in MBS;
2.  $S \leftarrow$  The last character in MBS ;
3.  $RS \leftarrow$  right characters of S in MathSymbol list ,
4.  $HRS \leftarrow S_i(S \text{ with } S_i \text{ is left level and } S_i \in RS)$ ;
5. If HRS is empty, return;
- 6  $MBS \leftarrow \text{FindStartSymbol}(HRS)$ ;
7. Put MBS into the last part of the list MBS;
8. GetMBS(MBS , MathSymbol) , namely, using recursion to get all main characters in the first layer.

According to the above expression, we can get the main characters of the first layer which are the base to do the structure analysis later. See figure 7:

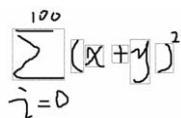


Figure 7. The main characters found by GetMBS()

**C. Identify every character in every fractal**

In general, the distance between a character in the mathematical expression and its fractal internal character

is closer than that of the external character, so Matsakis [13] and Tapis [17] suggest to using MST to solve the segmentation and structure analysis separately, but when utilizing MST, Tapia emphasizes excessively to use character dominating, meanwhile, the definition of attracting point is too tedious, and does not fit for the most expression, and in the analysis of expression structure, he didn't consider the logic structures of the expression. In order to better adapt to general mathematical expression structure analysis, we redefine the calculation method of the weight of the edge, as follows:

1) *Get the MST:* Some mature calculation methods of getting MST have already appeared, such as calculation method of Kruskal and calculation method of Prim, and so on. This paper adopted the Prim to get MST, the process of calculating will not be described again. After constructing the MST, collection T should include all edges of MST.

If we regard every character as a vertex of a diagram, all vertex will form a completely connection diagram without direction. The key problem is how to determine the weight of each edge in the diagram.

In the mathematical expression, the main characters in the first layer decide the main frame of the expression, generally speaking, the main characters in the first layer and the characters in their own shape can form a sub-tree, but because of the rule in handwriting expression, if we don't choose weight appropriately, the characters in one shape combining with the characters in another shape may form a sub-tree. See figure 8:

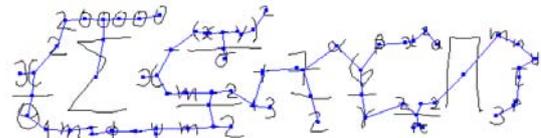


Figure 8. wrong MST basing on the centroid distance

We improved the calculation method of weight:

Making characters A and B' largest and smallest x ,y coordinate as  $A_{maxx}$  ,  $A_{maxy}$  ,  $A_{minx}$  ,  $A_{miny}$  ,  $B_{maxx}$  ,  $B_{maxy}$  ,  $B_{minx}$  ,  $B_{miny}$  , the calculation of weight  $S_{AB}$  of vertex A and B should follow the following rules[21]:

Rule 1:

If the relationship between A and B is dominating, then:

- (1) if the relationship between A and B is up,  $S_{AB} = (B_{miny} - A_{maxy})/2$
- (2) if the relationship between A and B is down,  $S_{AB} = (A_{miny} - B_{maxy})/2$
- (3) if the relationship between A and B is left-up, left level, and left-down, make  $S_1 = A_{minx} - B_{maxx}$  ,

$$S_2 = \begin{cases} B_{miny} - A_{maxy} & \text{AB is left-up relation} \\ A_{miny} - B_{maxy} & \text{AB is left-down relation} \\ 0 & \text{AB is left level relation} \end{cases}$$

So  $S_{AB} = \max(S_1, S_2)$ .

- (4) if the relationship between A and B is right-up, right level and right-down, make  $S_1 = B_{minx} - A_{maxx}$  ,

$$S_2 = \begin{cases} B_{\min y} - A_{\max y} & \text{AB is right-up relation} \\ A_{\min y} - B_{\max y} & \text{AB is right-down relation} \\ 0 & \text{AB is right level relation} \end{cases}$$

So  $S_{AB} = \max(S_1, S_2)$ .

Rule 2:

If the relationship between A and B is not dominating, then  $S_{AB} = \infty$

Rule 3:

If  $S_{AB} < 0, S_{AB} < 0$ .

Generally speaking, the objective characters who form up-down relationship with the subjective characters, if they form a shape, the distance is near, otherwise, the distance is far. So in the weight definition of every edge, when two characters own up-down relationship, dividing  $S_{AB}$  by 2, doing so just because, if two characters form a shape, the reducing weight will take the two as one edge of MST more effectively; if not, because of far distance, dividing by 2 will not influence the edge forming of MST in certain extent.

Because the main characters are the main structure of the expression, taking these main characters as the given vertex of MST, making  $V = S_1, S_2, \dots, S_t$ , their edge are  $E = [(S_1, S_2), (S_2, S_3), (S_{t-1}, S_t)]$ , when the other character nodes  $S_i$  and  $S_r (S_r \in V)$  weight are minimal, adding vertex  $S_i$  into  $V$ , size  $(S_r, S_i)$  into  $E$ , until all character nodes are added into  $V$ , forming a expression characters MST, see figure 9:

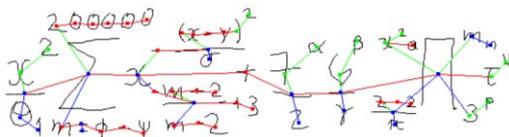


Figure 9. MST constructed by the method illustrated in this paper

The result of the experiment shows that the MST constructed by new rules can eliminate many shortcomings in the aspect of structure analysis.

2) *Determining table structure*: In the determined shape, there are some problems of table structure which can not be processed as a expression, or will produce some serious errors [10, 22], see figure 10:

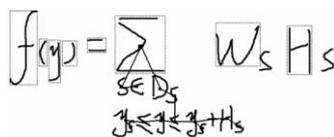


Figure 10. Table structure expression

By the statistical analysis to mathematical expression, we summarize several characteristics of table structure as following:

(1) The expression structure is simple, the main body of shape generally will not be fraction line, table structure will not form left-up, left-down, right-up or right-down with main body.

(2) When the relationship between characters is up or down, weight will be small, if a character is changeable, the distance is a little bit far.

(3) One expression in the table structure, the two characters with left-up, left-down, right-up, right-down relationships, will overlap in the plumb direction.

According to these characteristics, we add the up and down characters of fraction line, and right-up, right-down, left-up, left-down, as a expression, into the expression list Expression.

When the root node of tree is not fraction line, we judge whether the up or down characters are table structure by using Recognition\_Table(). We order the characters in the shape according to minimal x coordinate, forming character list  $T\_List = (S_1, S_2, \dots, S_n)$ , Algorithm is described as follows:

Recognition\_Table(T\_List):

1. If the root node character of collection is fraction line, return;

2. adding all objective characters  $S_k (S_k \in T\_List)$  from the sub-tree into Table\_List;

3. Every node in the Table\_List, if the child-node and parent-node meet one of the following two conditions, adding the child-node into Table\_List:

a) Overlapping in the vertical direction.

b) no overlapping in the vertical direction, but the space of besieging box in the vertical direction less than one value  $\delta$ . ( $\delta = H/8$ ,  $H$  is the height of the besieging box of the parent-node)

## VII. THE TEST RESULTS ON PC

We experimented on theoretical analysis, we found various expression 412 in many books. There are all kinds of shape, including table structure expression 60, character samples 26617, which construct a sample bank. In order to validate different writing habits influence the system, we choose 11 undergraduates to write, Han nationality 5, Hui, Uygur, Kazakstan, 2 individually. In the beginning, we taught them how to use this system. First, we introduce the system to them, when they enter 10 expressions, we will provide help when they need. After several exercise, ask them to enter 50 testing expressions which are constituted by 729 mathematical symbols and characters without any help, testing results are as following table III:

TABLE III. TEST RESULTS

test No.	Input characters	error	error rate (%)	Input expression	error	error rate (%)
1	729	6	0.82	50	3	6
2	729	7	0.96	50	3	6
3	729	19	2.60	50	6	12
4	729	16	2.19	50	5	10
5	729	3	0.41	50	1	2
6	729	28	3.84	50	8	16
7	729	14	1.92	50	3	6
8	729	2	0.27	50	2	4
9	729	5	0.69	50	2	4
10	729	5	0.69	50	3	6
11	729	1	0.14	50	1	2

VIII. THE CONSTRUCTION OF THE EMBEDDED ENVIRONMENT

This paper has contrived an online handwriting recognizing system of mathematical expression based on embedded equipment, which supports in the use of the languages of the minority, such as Uyghur and Kazakh languages, see figure 11. For the features of mathematical expression recognizing, and from the point of ensuring the system's efficiency, this paper has constructed an embedded platform, on which the capability of the system has been tested, and it has compared the data gained from the test with those in PC, which offers a gist to improve the algorithm and perfect the system.



Figure 11. Uyghur writing

Because it has different recognizing characteristics from Chinese characters and English letters and the users have various practical demands, the design for the embedded handwriting recognizing system of online mathematical expression should choose and apply an appropriate embedded system frame, such as the performance of the CPU, the size of RAM, the memorizer, the touch panel, the interface and the like. Then the appropriate algorithm can be chose and optimized to get the best effects.

In an embedded environment, the disposing process can be simply described as follows: the system gets the information of strokes first in dynamic by the touch screen, and transmits data to the memory of the system. Then the CPU gains the result of recognizing by operation and finally displays the recognized messages on LCD.

The details about the embedded platform of the recognizing system are as follows:

The Overall Frame of the Embedded System:

The figure 12 and the figure 13 list respectively the software components of the embedded system[23] and the frame of system functions.

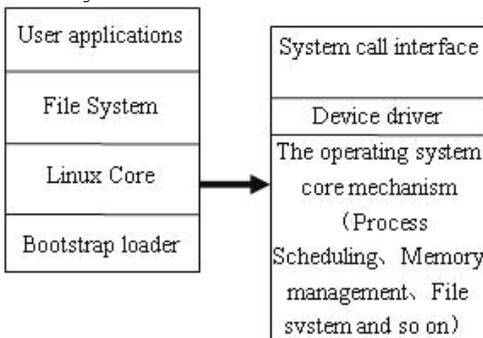


Figure 12. The software components of the embedded system

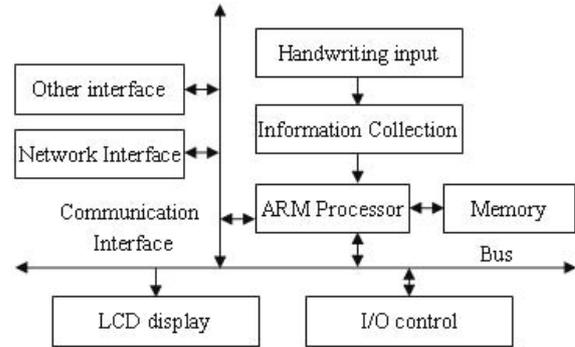


Figure 13. The frame of system functions

The aim of the research is clear and fixed and the frame of the function is made. After that, the step is to make a concrete design for the system.

A. Choosing CPU

It is a huge work to develop various drivers and manage its communications independently, which needs a long development cycle. Thus we can choose a type of operating system with drivers of all kinds to supervise and coordinate every software module. What we need do is just to make a simple transplanted, which greatly reduces the time of developing software. However, the operating system means the demand of CPU' high rate and the system's large memory. Luckily, the popular processor ARM with 4G address space at the market can meet the need of the memory that the loading operating system demands. Its 32 bits RISC feature can also meet the need the system speed requires. And moreover, the processor ARM is of rich resource with some primary interface controller, which makes it convenient to form a system[24].

ARM is the world-leading supplier of 16/32 bits embedded microprocessor solutions. The technique of the processor ARM, broadly used in the fields of portable communicate products, PDA, multimedia and embedded, has been the standard of RISC. The ARM processor is the engine of the system, which reads and carries out instructions of ARM(32bits) or Thumb(16bits) from the memory.

At present, popular processors are ARM7 and ARM9, the typical operating frequency being 60M (<133M) and 200M (<266M) respectively. Considering the high requirement of the whole system's real-time and the need of follow-up developing, we choose the processor ARM9 whose speed is quicker. It can run the operating systems, like Windows CE or Linux and process the complicated messages without difficulties. This paper chooses Samsung S3C2410 processor of high performance and low energy-consuming for the designed system, which is made of three parts, ARM9TDMI, MMU, and high-speed CACHE. The main resources attached to the operating system are: a LCD controller (a LCD screen with a touch panel, which supports SIN and TFT), a SDRAM controller, three accesses UATR, four accesses DMA, ten bits ADC of eight accesses, two USB host computer interface, a USB equipment interface, a watchdog timer, 117 bits general I/O and interface, 24 bits outer interrupt

sources and the like, which can fully meet the need of the designed system in this paper.

*B. The Construction of Storage System*

1) *FLASH memory*: The FLASH memory has the capacity of preserving data when the power supply is cut off. It is mainly used to store the solidified start-up codes, the core of operating system, file system, and initialization parameters. Presently, the fashionable FLASH memories at the market are NOR FLASH and NAND FLASH. NOR FLASH, the same with SDRAM, uses the memory random read technology, which makes it feasible to carry out directly the program stored in FLASH. While NAND is different, it reads a whole block of memory per time. Thus the program stored in NAND FLASH can not be run directly, which must be copied first in SDRAM and then can be run. But NAND structure can offer cells of high density to reach the standard of storage of high density, so its price is lower than that of NOR FLASH. Therefore, two alternatives of schemes here are supplied: One is to adopt NOR FLASH + NAND FLASH memory mode. NOR FLASH stores Bootloader which can be performed and has a small number of codes and some essential data. And NAND FLASH maintains the system core and file system with much large reserves. The other is as follows. According to the special NAND start-up performance of 53C2410, users can store guide- codes and operating system mirror in NAND FLASH outside of 53C2410 and start in NAND FLASH. When the processor is reset in this mode, inner-equipped NAND FLASH will load guide-codes automatically into the inner part SDRAM and then running. After that, the guide -program in SRAM will load the operating system mirror to SDRAM so that the operating system can be run in SDRAM, which can save the cost. The designed system in this paper adopts the first scheme, which can efficiently improve the operation efficiency.

2) *SDRAM memory*: Compared with the FLASH memory, SDRAM does not have the feature of preserving data when the power source is cut off, but its speed of storing and reading is much higher than that of the FLASH memory and it has the attribute of reading and writing. Thus, SDRAM in the system is chiefly used as the program run spaces, data area and stack area. When the system is started, CPU reads the start-up codes from the reset address first. After system initialization, program codes are transferred into SDRAM to run, which can pick up the operation speed of the system. Meanwhile, the system codes and users stack, running data can be put into SDRAM. The memory cell in SDRAM can be treated as a capacitance, which always has a tendency to discharge. The way to avoid losing data is to refresh (charge) on time. Therefore, if the SDRAM is used in the system, the microprocessor is demanded to possess the logic of refresh control, or the logic circuit of refresh control should be added in other place of the system. 53C2410 has an independent logic of SDRAM refresh

control, which makes it convenient to interface with SDRAM[25].

*C. LCD*

When developing the embedded products, in addition to the operating system, the chart support system is another one of systematical software modules which are received the most attention. As long as the embedded products are faced with human-computer interaction, the problems of outputting characters or charts are involved. The embedded recognizing system designed in this paper has a friendly GUI alternation interface that makes users to operate conveniently, and it adopts MiniGUI to be the library of the interface development. As the medium software between the operating system and the application, MiniGUI hides the differences between the operating system and the hardware platform, supplying consistent functions and capacities to the upper level—the application[25].

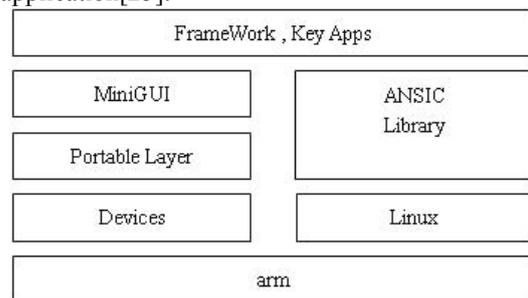


Figure 14. The location of MiniGUI at the software model

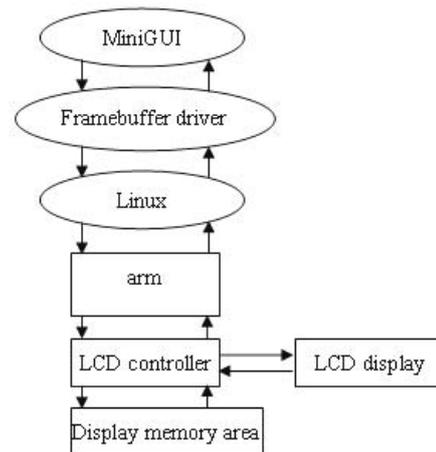


Figure 15. The role of MiniGUI at the display system

*D. The Operating system*

For the moment, products of the embedded operating system, popular at the market, are of great variety, such as windows CE, Vxwork,  $\mu$ C/OS, Linux, and so on. Windows CE is a kind of 32 bits multi-tasking operating system. It can develop various equipments suitable for companies and clients after it is compressed and transplanted. Because Windows CE is the center of the Microsoft's "The Venus Plan", containing Internet Explorer edition, which can realize connecting with the Internet and synchronously exchanging messages. Good applications can be developed on the basis of Windows

CE, but the price of windows CE is much high. Besides the expensive charge of copyright, extra costs are charged for the use of each product applying the operating system. Therefore the cost of the products is much high[24].

Vxwork contains the operating system itself and its integrated development environment. Vxwork uses the structure of microkernel, possessing the characteristics of supporting many types of processors, various network protocol, good compatibility and feature of cutting, and the functions of dynamic link and download. It was Vxwork that was adopted by "The Chance" which landed on the Mars in 2004. Vxwork gains a great number of users because of its reliability, openness, real-time and friendly, but it charges expensive copyright fees and use fees.

$\mu$  C/OS is suitable for a miniature controlling system, which performs with high efficiency, uses a little space, and possesses the qualities of good real-time and better expansibility. The smallest core codes can be translated to 2KB. It is just the start of the task in designing system to transplant  $\mu$ C/OS into the platform of target hardware. The next step is to expand functions in accordance with the practical application needs towards the operating systems, including a rockbottom hardware drivers, file systems, GUI etc., to build a practical RTOS. However these follow-up developments are not expected.

As a server operating system which is a free and open source code, Linux has aroused an abroad attention in business circles with its functions growing better. Linux is gradually applied on the workstation and desktop PC. Meantime, it has become an ideal operating system for the intellectual embedded facility on account of the characteristic of its server OS. At present, no matter what the aim is to study the operation system Linux itself or to study the embedded Linux, there are many corps for development on the Internet and we can find a variety of corresponding programs and documents.

With an overall consideration towards the features of these operating systems and the aim to minimize the cost of the whole system, we choose the Linux to be the operating system of the designed system in this paper.

#### E. Network interface

As a type of embedded handwriting recognizing system, its basic task is to recognize and display mathematical expressions. Then it can do simple operation and show us the results on the embedded facility are feasible. However with the limits of the embedded facility's hardware environment, it is impossible for the system to do complicated operations (such as complicated calculus and matrix). Because doing complicated operations depends on using professional mathematical software, thus if the embedded handwriting recognizing system can be connected with a server by the means of wireless to exchange the necessary data message, it can achieve the complicated mathematical operations, which undoubtedly will increase its own use value. Therefore for the whole system and the need to expand follow-up systems, Ethernet interface circuit is essential, but meantime relatively complicated. From the view of the hardware, Ethernet interface circuit chiefly

consists of two parts: the controller MAC and the physical layer interface. Presently, it is common to find Ethernet interface circuit as these: RTL8019、TRL8029、TRL8039、CS8900、DM9008 etc., whose inner parts also contain the two parts mentioned above. 53C2410, embedded with a Ethernet controller, supports medium independent interface and DMA interface with buffer. And it realizes connecting with the Internet in the mode of half-duplex or full-duplex. .

53C2410 whose inner part contains an Ethernet controller does not offer physical layer interface, so it need connect a physical layer chip on the outside of 53C2410 to supply a channel for the Ethernet. Because the system described in this paper has not realized the communication functions of the wireless network, it is not necessary to talk about it.

#### F. Target board environment

As Linux is the basis of software for running the system, whether the construction of the system is success depends on the correct running of the Linux. What we should do in order to make the system run automatically after boot-strap is to solidify the transplanted codes on the address 0 of the memorizer, so that CPU will begin to carry out the first instruction from the address 0 after the computer is electrified. The figure is a target board environment when the system is starting up.



Figure 16. Target board desktop environment

### IX. COMPARISON AND ANALYSIS OF THE RESULTS

The performance of the target board recognition system was tested using the same test method with the PC environment in order to facilitate comparison. Testing results are as following table IV:

TABLE IV. THE RESULTS (ERROR AND ERROR RATE ARE ALL AVERAGE)

Test environment	Input characters	error	error rate (%)	Input expression	error	error rate (%)
PC	729	9.63	1.32	50	3.36	6.72
Target board	729	13.12	1.80	50	7.22	14.44

(1)The expression refusing reorganization rate listed in table IV is the testing results without any correcting

after entering, through stroke correcting, character recognizing correcting, revising the expression structure, all subjects can give 50 testing expressions in the last system correct recognizing results.

(2) Part of the entered expressions which can not be analyzed structure successfully, will lead to expression refusing recognizing; the reason is that the place arrangement of entered expressions is illogical. Reducing the refusing recognizing rate, improving the capability of expression editor is the key work in the further research. Meanwhile, when the expression structure were misunderstanding the enter material, we try to make the system tell the users automatically, and comb-out the circle of expression debar barrier which do not give any clue to users.

(3) Because of the adding of various structures, the ability of recognizing will low down, but comparing with the former researches, the method in this paper heightens the correct recognizing rate of the expressions obviously, which show that this method is feasible.

(4) The current recognizing algorithm is developed on the platform Visual C++ of PC. Thus it should be properly modified continuously so that it can run well in the frame of the Linux + MiniGUI in this system to get better effects, which is the important task done in the future research.

#### REFERENCES

- [1] Yusup, etc. Handwriting Uighur character recognition and electronic document editing technology development[R], Xinjiang Normal University, 2007.
- [2] Kam Fai Chan, etc. An efficient syntactic approach to structural analysis of online handwritten mathematical expressions. *Pattern Recognition* 33 (3), Mar 2000:375-384.
- [3] K Chan, D Yeung. Mathematics expression recognition: A Survey, *Intl. Journal on Document Analysis and Recognition*, 3(1), August 2000:3-15.
- [4] Zanibbi, D Blostein, J cordy. Recognizing mathematical expressions using tree transformation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(11), Nov 2002.
- [5] Tapia, R Rojas. Recognition of online handwritten mathematical expressions using a minimum spanning tree construction and symbol dominance. In *Proceedings of the Fifth IAPR International Workshop on Graphics Recognition (GREC)*, Barcelona, Spain, July, 2003:329-340.
- [6] Shi Kuo Chang. A Method for the structural analysis of two-dimensional expressions information Sciences, 1970.
- [7] Hong Liurong, etc. Recognition of Spatial Relationships between Symbols or Characters in Mathematical Expressions. *Computer Engineering and Applications*, 2006.
- [8] R Zanibbi. Recognition of mathematics notation via computer using baseline structure. Technical Report ISBN-0836-0227-439. Dept. Computing and Information Science, Queen's University, Kingston, Ontario, August.
- [9] R H Anderson. Syntax-directed recognition of handprinted two-dimensional mathematics. In: M Klerer, J Reinfelds *Interactive Systems for Experimental Applied Mathematics*, 436-459. Academic Press, New York, 1968.
- [10] ZHANG Jian-cheng, etc. A Survey on Recognition of On-line Handwritten Mathematical Expression, *Journal of Huaibei Coal Industry Teachers College*, 2008.
- [11] William. Computer input/output of mathematical expressions. In *proceeding of the Second Symposium on symbolic and Algebraic Manipulation*, March 1971:78-89.
- [12] Xiao Jianyu, Wang Qianping, Hong Liurong. Recognition of mathematical expressions based on convex hull and fuzzy recognition. *Computer Applications and Software*, 2008.
- [13] N. Matsakis. Recognition of Handwritten Mathematical Expressions. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, May 1999.
- [14] XIAO Jian-yu, etc. Determine superscript and subscript relations in mathematical expressions based on fuzzy theory. *Computer Engineering and Design*. 2008.
- [15] Hong Liurong, Wang Yaocai. Handwritten Numeral Recognition Using Graph Theory and Direction Information. *Computer Engineering*. 2006.
- [16] G. Friedland, L. Knipping, and R. Rojas. E-Chalk Technical Description. Technical Report B-02-11, Freie Universit"at Berlin, Institut f"ur Informatik, 2002.
- [17] E. Tapia, etc. Recognition of On-Line Handwritten Mathematical Expressions in the E-Chalk System. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR)*, August 2003.
- [18] E. Tapia. JMathNotes: A Java-Based Editor for On-Line Handwritten Mathematical Expressions, January 2004.
- [19] D. Blostein, etc. Recognition of Mathematical Notation. In P. S. P. Wang and H. Bunke, editors, *Handbook on Optical Character Recognition and Document Analysis*. World Scientific Publishing Company, 1996.
- [20] H. J. Lee and J. S. Wang. Design of a Mathematical Expression Understanding System. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR)*, pages 1084-1087, 1995.
- [21] Hong Liurong, Study for the key technology of the on-line handwritten mathematical formula recognition (M), 2007.
- [22] Kam-Fai Chan, etc. Mathematical Expression Recognition: a Survey. *International Journal on Document Analysis and Recognition (IJ DAR)*, 3(1):3-15, 2000.
- [23] Li Lina, Embedded System Status and Development, *Journal of Changchun Industrial University*, 2004
- [24] The Intelligent Transportation Society of America. <http://www.itsa.org>
- [25] Wei Yongming, etc. Embedded software development and C language realize, *Electronics Industry Press*, 2008.

**Qi Xiangwei** Living in Urumqi, China, received his B.Sc. degree from Liaocheng University, study Master's degree at Xinjiang Normal University now; His major research interests include Chinese computing, human-computer interaction and pattern recognition.

**Yusup Abaydulla** Living in Urumqi, China, Professor, and His major research interests include Chinese computing, human-computer interaction and pattern recognition.