A SYMBOL LAYOUT CLASSIFICATION FOR MATHEMATICAL FORMULA USING LAYOUT CONTEXT

by

Ling Ouyang

B.S., Huazhong University of Science and Technology, China, 2006

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Chester F. Carlson Center for Imaging Science
of the College of Science
Rochester Institute of Technology

Nov 17, 2009

Signature of the Author ______________________________

Accepted by ______________________________
Coordinator, M.S. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Ling Ouyang
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

Dr. Richard Zanibbi, Thesis Advisor

Dr. Carl Salvaggio

Dr. Roger Easton

Date

THESIS RELEASE PERMISSION

ROCHESTER INSTITUTE OF TECHNOLOGY

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Thesis:

A SYMBOL LAYOUT CLASSIFICATION FOR MATHEMATICAL FORMULA USING LAYOUT CONTEXT

I, Ling Ouyang, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature ______________________________

Date

Acknowledgments

Thanks to Dr. Richard Zanibbi for providing me with an excellent guidance and supervision. I really have greatly appreciate his patience, creativity and enthusiasm when while we worked together over the past one year. Without his instruction, I would have not been able to accomplish my research work.

I was also lucky to have two other committee members, Dr. Roger Easton and Dr. Carl Salvaggio, available to assist me. They offered many invaluable suggestions during my research work and thesis writing.

I also would also like to thank Amit and Li for the their discussions about my research work. It was a great pleasure to work with them in DPRL.

In addition, I am thankful for the love and support of my family, especially my parents Xiao Ouyang and Lifang Yuan, my grandfather Shitao Ouyang, my aunts Jun Ouyang, Hong Ouyang and my uncles Xiaofan Feng and Mark Jackson.

At last, I appreciate my best friend Fan for her love and encouragement during my thesis writing and I would like to dedicate this thesis to her.

A SYMBOL LAYOUT CLASSIFICATION FOR MATHEMATICAL FORMULA USING LAYOUT CONTEXT

Publication No. ____________________

Ling Ouyang, M.S.
Rochester Institute of Technology, 2009

Supervisor: Richard Zanibbi

Abstract

We describe a symbol classification technique for identifying the expected locations of neighboring symbols in mathematical expressions. We use the seven symbol layout classes of the DRACULAE math notation parser (Zanibbi, et al., 2002) to represent expected locations for neighboring symbols: Ascender, Descender, Centered, Open Bracket, Non-Scripted, Variable Range (e.g., integrals) and Root. A new feature based on the shape context (Belongie, et al., 2002), named *layout context*, is used to describe the arrangement of neighboring symbols relative to a reference symbol, and the nearest neighbor rule is used for classification. 1917 mathematical symbols from the University of Washington III document database are used in our experiments. Using a leave-one-out estimate, our best classification rate reaches nearly 80%. In our experiments, we find that the size of the reference symbol neighborhood area, the number and the sampling positions of the points of the key points model representing a symbol's location, play important roles in the classification process.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

The recognition of mathematical formulas is a challenging pattern recognition problem due to the large number of math symbols and complexities in interpreting the two-dimensional arrangement of symbols and their intended semantics. There are usually two processes involved in a mathematical formula recognition system : symbol recognition and structure analysis [6]. Symbol recognition is the process of identifying the identity while structure analysis is intended to determine the spatial and logical relationships between the mathematical symbols of an expression. This spatial relation is particularly important since much of the information in mathematical expressions is carried by the relative spatial position between symbols, such as superscript, subscript, adjacent and containment (e.g., in a square root). Many methods have been proposed to extract spatial relationships between symbols, such as coordinate grammars [4], projection profile cutting [7], minimum spanning trees for penalty graphs representing alternative symbol layouts [2], recursive baseline structure analysis [1], and others.

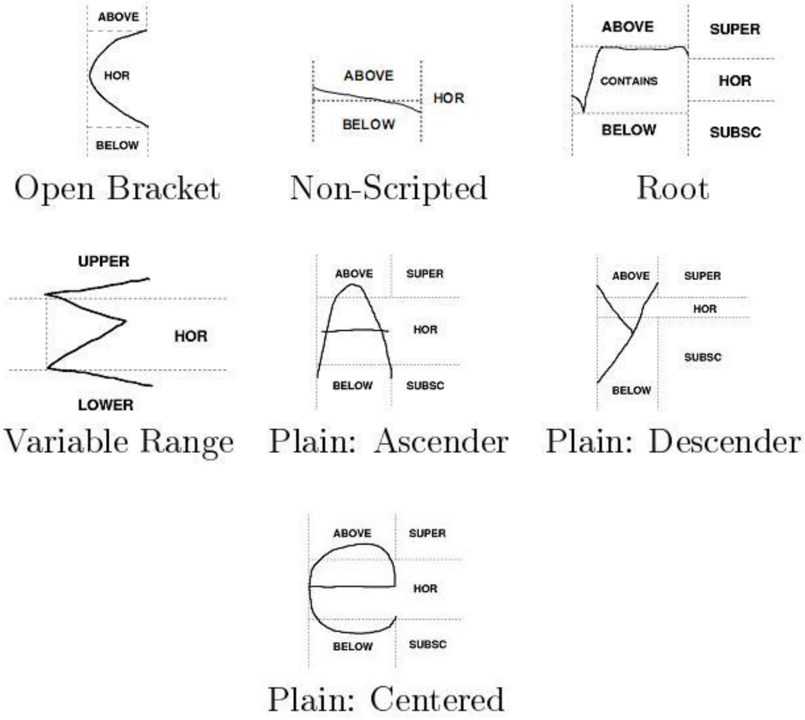
To explore the spatial relationship between a mathematical symbol and its neighboring symbols within the expression, we present an algorithm for classifying all mathematical symbols into seven layout classes (Table 1.1), which identify the expected locations of neighboring symbols with a significant spatial relationship. From these seven classes, existing techniques (in particular, baseline structure analysis [1]) may be used to identify the spatial arrangement of symbols in an expression. Surrounding regions of a symbol may include above, below, superscript, subscript, horizontal adjacency and containment. Different classes have different associated regions, as shown in Figure 1.1.

Table 1.1: Class Membership [1]

Class	**Symbols**
Ascender	$0...9, A...Z, b, d, f, h, i, k, l, t,$ $\Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi$
Descender	$g, p, q, y, \gamma, \eta, \mu, \rho, \chi, \psi$
Open Bracket	[{(
Non-Scripted	Unary binary operators and relation $(\times, \backslash, \geq, \div, \equiv)$
Root	$\sqrt{\ }$
Variable Range	$\sum \prod \int \bigcap \bigcup$
Center	All other symbols

We use a feature named *layout context* to describe the arrangement of neighboring symbols relative to a reference symbol. A number of key points sampled from the side and/or interior of the symbol bounding box are used to represent symbol locations. A circle placed at the reference bounding box center with adjustable radius is used to segment the neighboring symbol region. We examine a variety of key point models and neighborhood sizes, and use the Nearest Neighbor (NN) rule for classification. Depending on the chosen parameters, we have obtained a classification accuracy between 43% and 79.2% on symbols taken from math expressions in the University of Washington III document database.

Thesis Statement: One can increase the accuracy of symbol layout classification using layout contexts, by using a large number of key points from both the inside and the boundary of the symbol bounding box, and a small neighborhood size. The symbol bounding box is defined as the smallest rectangle that contains all pixels of the symbol image. The neighborhood refers to a circular area centered at the symbol bounding box center with a radius equal or larger than half of the length of the symbol bounding box diagonal.



Open Bracket

Non-Scripted

Root

Variable Range

Plain: Ascender

Plain: Descender

Plain: Centered

Figure 1.1: Symbol layout classes [1].

1.1 Limitation and Assumptions

Several limitations are acknowledged in our proposed classification method.

1. We have only used seven layout classes in the DRACULAE model [1] to explore the spatial relationships between the symbols within the mathematical expressions

2. All mathematical symbols are within expressions. Isolated symbols which exist in the text line are not considered in our research. For example, in the text line "the value of x can be figured out by solving the function $x^2+6x = 9$", the first x between the word "of" and the word "can" cannot be assigned any layout class since this symbol is isolated by the text letters. However we can classify the second x and the third x into proper layout classes because they are within the math expression.

3. The reference symbol and its neighboring symbols are from the same expression. Symbols from other expressions may not be identified as neighbors of the reference one. Suppose there are two mathematical expressions in the same document images, which are $x + y = z$ and $t^2 - 6 = u$. Then the neighboring symbols of x in the first expression can only be selected from the symbols within the first expression, such as $+$, y, $=$ and z. Any symbols in the second expression, such as t, 2, $-$, 6, $=$ and u, cannot be counted as the neighboring symbols of x in the first expression $x + y = z$.

4. We only focus on printed mathematical expressions. Handwritten expressions are not covered in our work.

5. All the symbols have been segmented and attributed with their bounding box coordinates. For example, if two symbols are touched together in the expression, they

would not be taken into account in our work unless they are well segmented. In addition, all the symbols' bounding box locations in the document images are given.

1.2 Contribution

1. An overall symbol layout classification accuracy of nearly 80% has been achieved using a total number of 1917 symbols from 73 expressions.

2. A new feature, named *layout context*, has been defined to describe the layout information of a symbol within the scope of the expression to which the symbol belongs.

3. Experiments have been performed to test the usefulness of the layout context feature in identifying symbol layout classes for mathematical expressions at different parameterizations of the feature. It is found that the best classification results are obtained if using a small circular neighborhood area that includes the closest surrounding symbols and a key point model with its points sampled from the sides and interior parts of the bounding box of both the reference symbol and the neighboring symbols.

Chapter 2

Background

To fully understand the motivation for the research topic and the inspiration for the methods developed in this paper, some important previous works are reviewed in this chapter.

2.1 Mathematical Formula Recognition

Mathematical formula recognition involves two main phases: symbol recognition and structural analysis. Symbol recognition recognizes the identity of the mathematical symbol in the expression. Structure analysis determines the spatial and logical relationship between the mathematic symbols within the formula. The two main activities (symbol recognition and structure analysis) exist in all mathematical formula recognition systems. Each may be subdivided into more specific processes:

Symbol recognition:

- Preprocessing
- Segmentation
- Recognition of symbols

Structural analysis:

- Symbol layout analysis
- Syntax and semantic analysis

The block diagram of a typical mathematical formula recognition system is shown in Figure 2.1. The input to the system is a scanned document image that includes both mathematical formulas and non-mathematical contents. After preprocessing, the image regions that only contain mathematical formulas are separated from the original document image. In the segmentation step, the images of mathematical formulas are decomposed into sub-images of individual symbols. Then the locations of these isolated symbols in the mathematical formula image are identified and the identities of the symbols are labeled in the step of recognition of symbols. In the symbol layout analysis, the spatial arrangement of symbols in the formula are described by a proper layout model such as baseline tree [1] and virtual link network [2]. In addition, the compound symbols (e.g., sin) that consists of a sequence of symbols are grouped and the structural symbols (e.g., fraction, integral) are labeled [1] in the layout model. Finally the mathematical semantics of the formula are interpreted in the syntax and semantics analysis. The syntax analysis attempts to identify the logical relationships between symbols based on the layout (e.g., fractions) and semantic analysis determents what mathematical operations/contents these logical relationships represent (e.g., fractions represent division of the numerator term by the denominator term). A form that may be translated by a Computer Algebra System, such as Maple [8], is outputted in this step.

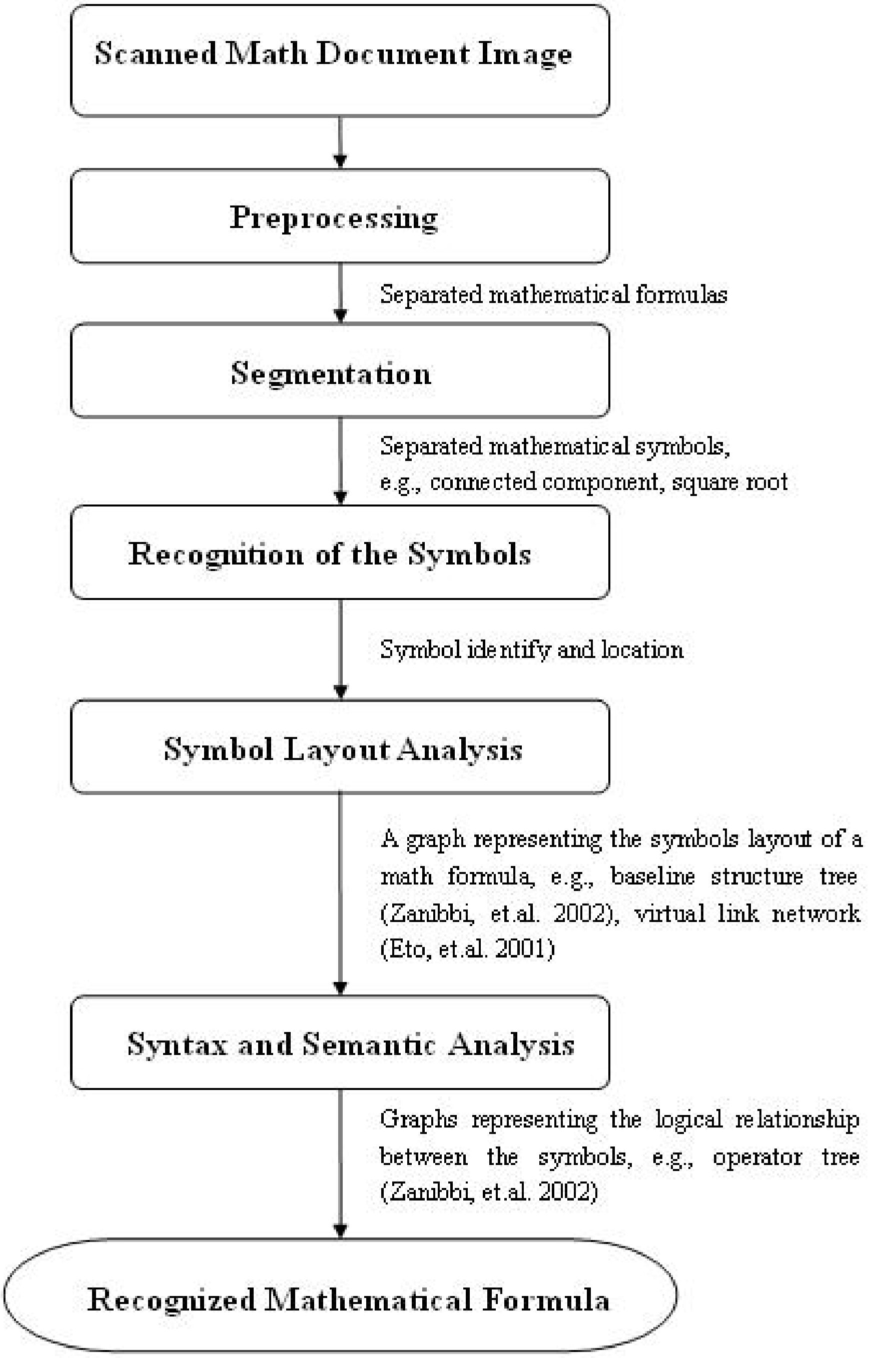

Scanned Math Document Image
Preprocessing
Separated mathematical formulas
Segmentation
Separated mathematical symbols,
e.g., connected component, square root
Recognition of the Symbols
Symbol identify and location
Symbol Layout Analysis
A graph representing the symbols layout of a math formula, e.g., baseline structure tree (Zanibbi, et.al. 2002), virtual link network (Eto, et.al. 2001)
Syntax and Semantic Analysis
Graphs representing the logical relationship between the symbols, e.g., operator tree (Zanibbi, et.al. 2002)
Recognized Mathematical Formula

Figure 2.1: A typical mathematical formula recognition system.

2.1.1 Symbol Recognition

Symbol recognition in mathematical formula recognition usually consists of three steps: preprocessing, segmentation, and recognition of symbols.

2.1.1.1 Preprocessing

Preprocessing provides “clean” input data that is easy to be used in the later recognition process. Some preprocessing operations, such as noise reduction (filtering, morphological operation), symbol normalization (skew normalization, baseline assignment) and data compression (binarization, thinning) are performed on the document page images. More details of the preprocessing techniques can be found in [9]. The output of this process are images regions that contain separated mathematical formulas.

2.1.1.2 Segmentation

Segmentation is the stage where the individual mathematical symbols are separated from the mathematical formulas. Okamoto, et al., [7] applied the projection profile cutting to partition a given printed expression into symbol blocks by using the blank spacing between rows or horizontally adjacent characters. This technique works quite well for separating symbols in a connected component form (e.g., $1, 2, 3, a, b$). However, for some symbols consisting of multiple components, such as i, j, % and =, a merging process is needed to combine the components for each symbol before they can be recognized. In addition, some symbols, such as square roots, usually contain other symbols inside their effective regions and thus generate more complexity for the segmentation. Faure and

Wang [10] used a mask removal operation to separate this kind of symbol and their embedded symbols before the projection profile cutting process is applied for the embedded symbols. Xue, et al., [11] proposed a connected component labeling method to segment a multiple symbols block, such as $\sqrt{\frac{\pi}{2}}$, into several small single symbol blocks after a coarse projection profile cutting operation. A more comprehensive survey of the character segmentation methods can be found in [12][13][14][15]. The output of this process are usually connected components.

2.1.1.3 Recognition of Symbols

Recognition of symbols is usually performed after segmentation to identify the individual symbol. Recognizing the isolated symbols usually consists of two steps: feature extraction and classification.

Common low-dimensional features include aspect ratio and crossing features; high-dimensional features include directional ones and peripheral ones. More details about the feature extraction methods are available in [16].

After selecting the proper features, classification assigns each symbol to a category. The supervised nonparametric classification methods are the ones that are most interested in our work and would be performed for the layout classification due to their two main characteristics :

1. The classification is based on the availability of the training data sets with a given category label.

2. There is no assumption of any probability density functions for the classifier.

Typical methods of the supervised nonparametric classification methods include Nearest Neighbor and K Nearest Neighbor. Nearest Neighbor rule is a method for classifying the object based on the class membership of the closest training sample in the feature space. The training samples are vectors in a multidimensional feature space with class labels. To measure the distances between the query sample and the training samples, some distance metrics, such as Euclidean distance, Mahalanobis distance, Hamming distance and χ^2 distance, are computed between the two sample points in the feature space. K Nearest Neighbor rule is a more general version of the Nearest Neighbor rule. Instead of predicting the test sample's class based on that of the closest training sample, a majority vote is applied among the k nearest training samples to determine the final class label for the test sample. One advantage of the Nearest Neighbor and K Nearest Neighbor rule is that it is very easy to implement these algorithms by just computing the distance from the test sample and the training samples. In addition, people may use these classification methods to study the strength of the feature since the efficiency of these methods is largely dependant on how well the feature separates the classes. The drawback to the Nearest Neighbor and K Nearest Neighbor is that the classification accuracy may be degraded by the noise data or the irrelevant feature elements in the feature vector. To reduce the effect of the noisy data, one can increase the k value to include more neighboring training samples.

2.1.2 Structural Analysis

Structural analysis is performed to build the hierarchical structure that describes the symbol spatial relationships and their semantic meaning. There are two main steps addressed in this phase: layout model construction and syntax and semantic analysis.

2.1.2.1 Symbol Layout Analysis

Due to the complexity of the two-dimensional spatial arrangement of symbols within mathematical expressions, different layout models are developed to describe the spatial relationships between the symbols.

Okamoto [17] defined normalized size and normalized center to describe the position relationship and size difference between adjacent symbols of an expression.

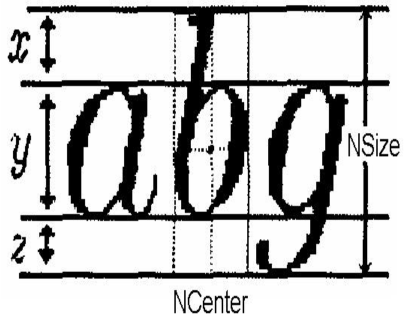


Figure 2.2: The normalized size (NSize) and normalized center (NCenter) of symbols with different baselines [2].

As illustrated in Figure 2.2, the normalized size is the total size (NSize) of a bounding box, including both the ascender part x, center part y and descender part z for symbols with different baselines. Normalized center is the related center of the bounding box. Then symbols that have similar normalized sizes and centers are extracted as one string based on the condition of size, condition of center and whether the symbols are contained in the same string. After that, a bottom-up method was proposed to analyze the layout structure for root expression, subscript/superscript expression.

Eto and Suzuki [2] proposed a layout model that represents the relation of a pair of mathematical symbols (parent and child) by a point in a normalized coordinate plane labeled by (H,D), which is defined based on the concept of normalized size and normalized center, as mentioned above. To illustrate the process of computing (H,D), an example is shown in Figure 2.3. For the pair of symbols in the expression X^2, h_1 and c_1 are the normalized sizes and normalized center of the parent character X, while h_2 and c_2 are the normalized size and normalized center of the child character 2. The formulas to calculate H and D are

$$H = \frac{h_1}{h_2} \cdot (1000) \tag{2.1}$$

$$D = \frac{c_1 - c_2}{h_1} \cdot (1000) \tag{2.2}$$

Using that pair of points, all sampled symbol pairs within the expressions are classified into four groups of characters types: Alphabets-Alphabets, Alphabets-Operators, Integrals-Alphabets and Big Operators-Alphabets. The area of the distribution plot of these different types of symbol pairs is used to calculate the cost of the link in the virtual link network, which connects the symbols within the formula with labels representing the possible relations of the pair of characters.

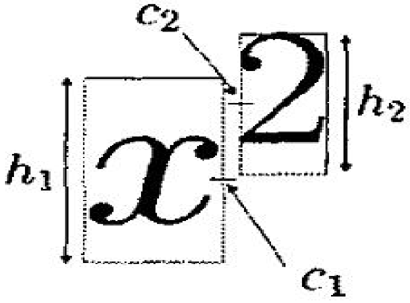


Figure 2.3: Definition of H and D using the normalized size $h1$ and $h2$ and normalized center $c1$ and $c2$ of the pair of symbols [2].

Chen proposed a structure pattern in terms of the relationships between operators and expressions [3]. These relationships include front superscript, front subscript, front, upper, inside, lower, back superscript, and back and up subscript, as shown in Figure 2.4. Mathematical rules are constructed using these relationships to describe various types of factors of the rules, such as the root factor, the matrix factor, the fraction factor and so on. By dividing a formula into horizontal groups from left to right, a layout tree, which consists of the types of the symbols, their positions, sizes, centerlines, and the parent-child relationship, is built to parse the formula.

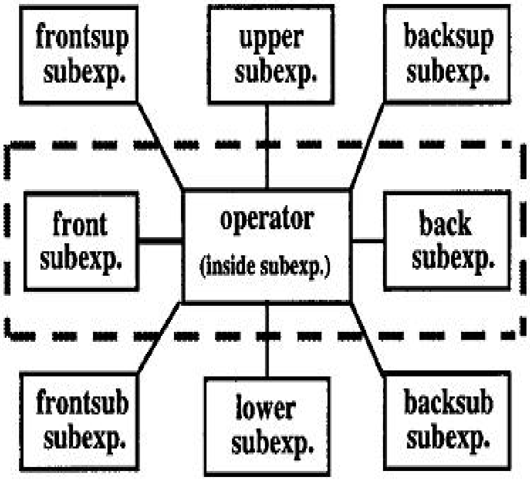


Figure 2.4: Spatial relationships between operators and surrounding subexpressions [3].

Anderson [4] represented mathematical symbols using a bounding box and its related corner coordinates, which are *xmin*, *xmax*, *ymin*, *ymax*. Two additional positional coordinates (*xcenter* and *ycenter*) describe the typographical center of the character as shown in Figure 2.5. The middle point between *xmin* and *xmax* is "xcenter" and "ycenter" is computed from *ymin* and *ymax* depending on the position of the symbol relative to the baseline. The typographical center is another form of the normalized center, which also reflects the relative position of the adjacent symbols of different types including ascending symbol, normal symbols and descending symbols [18].

Zanibbi [1] predefined seven symbol layout classes: Ascender, Descender, Center, Open Bracket, Non-Scripted, Variable Range and Root, as shown in Table 1.1. Different classes of symbols have different locations of surrounding regions. All these regions have important spatial relationship with the reference symbol (Figure 1.1). Two parameters,

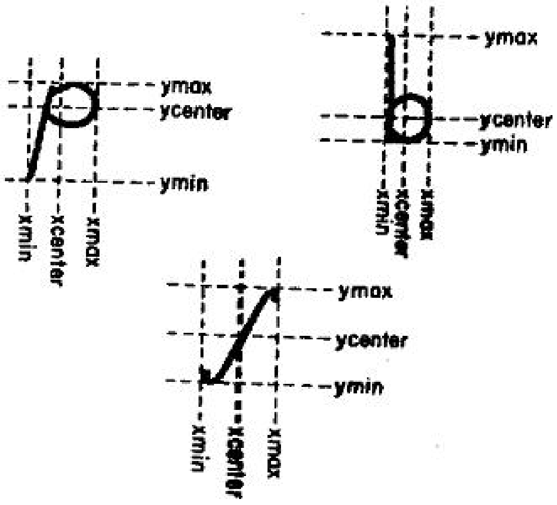


Figure 2.5: Typographical centroid and bounding box coordinates of a symbol [4].

centroid ratio c and threshold ratio t, describe the position of the region in terms of the symbol bounding box height. Using the layout model, a baseline structure tree (BST) is built to represent the spatial relationship between the symbols of a mathematical formula. Baseline is defined as a horizontal linear arrangement of the symbols. For example, there are two baselines in the expression $y^{5+b} - 4$. The baseline for "$5 + b$" is defined as the nested baseline in which the symbols are either contained by a symbol (such as square root) or vertically offset a symbol (such as a symbol's superscript or subscript). In this example formula, the baseline "$5 + b$" is nested on the right superscripted region relative to y. The baseline "$y - 4$" is defined as the dominant baseline in which the symbols are not nested relative to any symbols. After finding all baselines, a baseline structure tree is built and successively refined and restructured to represent the different types of baselines including both dominant baseline and nested baselines. To parse the expression in the later syntax and semantic analysis, two types of symbols are explicitly labeled. One is the compound symbol, such as sin, cos, which consists of multiple input symbols. The other one is the structure symbol, such as fractions, whose function depends on the structure between multiple baselines.

In general, most of the above layout models are constructed using the symbol typographical center position, the bounding box size of the symbol and location, which are crucial information to describe the spatial relationships between the symbols of different types and sizes within the expressions.

2.1.2.2 Syntax and Semantic Analysis

After representing the two-dimensional spatial arrangement by proper layout models, the sematic meanings based on these spatial structures are analyzed by using different methods as described below.

Okamoto [17] developed a two-way structure analysis method to analyze the semantic meaning of the mathematical expression structure . He found that, although the recursive projection-profile cutting [7] had good performance on the recognition of some mathematical formulas, some errors were caused by over-cutting in the very early recognition process and the contained structure of some symbols, such as root. To solve these problems, the bottom-up and the top-down strategies are applied in the structure analysis process. For basic structures, such as vertical and horizontal relations between subexpressions, the top-down strategy is used to find the largest horizontal bar of the fraction before the next largest in the subexpression in the numerator and denominator. For specific structures, such as root expressions, superscript and subscript expressions, and matrix expressions, the bottom-up strategy is applied. In this situation, the particular relations such as superscript, subscript, and root are processed and then the subexpression is treated as a single component in the later steps.

A pure top-down approach, which is syntax directed and guided by some grammar rules, is applied in Anderson's system [4]. The input of the system is an attributed symbol list consisting of the symbol identity, bounding box coordinates and typographical center coordinates. Then a parse tree is built by trying all possible partition symbols, and the semantics of the expression are obtained by transferring the string attributes from leaves to root of the tree.

A method to recognize the mathematical formulas by searching for spanning trees of the virtual link network with minimum costs that correspond to the recognition result of the mathematical formula structure was developed in the work of Eto and Suzuki [2]. Both the local cost initially attached to the network regarding to the relationship between adjacent symbols pair and the global cost reflecting the whole formula structure are included. The advantage of this method is that it is very robust to the recognition error between symbols of different normalized sizes.

Mathematical rules were developed by Chen [3] to automatically parse the layout structure and semantics for the mathematical notations. The factors of the rule are derived from the constructed layout tree model that describes the spatial relationships between the symbols and subexpressions within a mathematical formula.

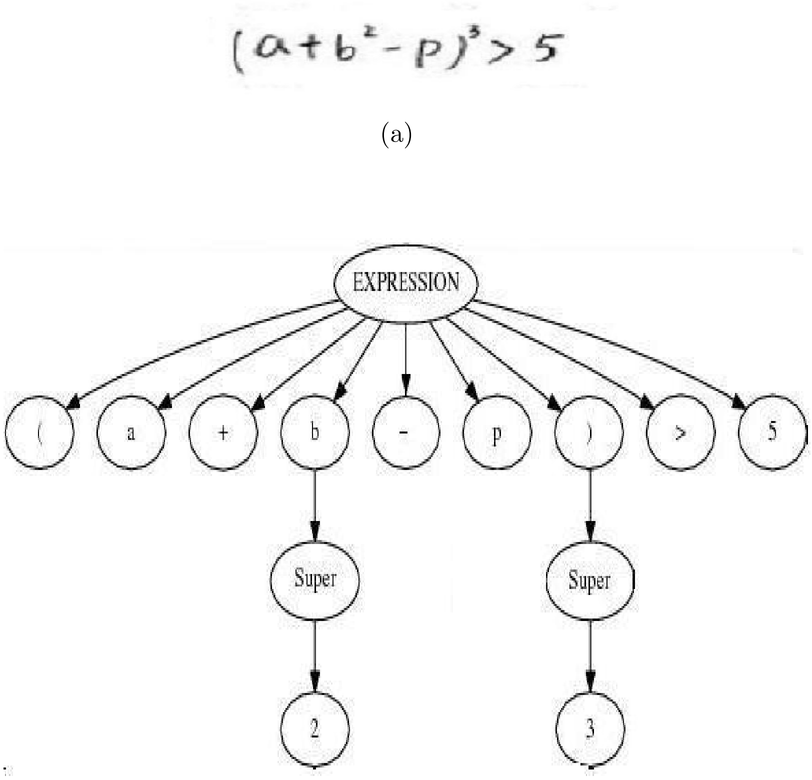
Zanibbi [1] proposed a method to analyze the syntax and semantics of an expression based on the baseline structure tree that represents the spatial relationships between the symbols. A mathematical formula grammar and a set of tree transformation are applied to create operator trees that encode all the information necessary to interpret the semantics of a mathematical formula. The grammars specify the precedence and associativity of the operators. The types of operands and the implied operators are recognized by using a set of tree transformation rules to identify these patterns in the formula parsing tree.

2.2 DRACULAE

An implementation called Diagram Recognition Application for Computer Understanding of Large Algebraic Expressions (DRACULAE) [1] is introduced. DRACULAE serves as an expression parser that performs structure analysis for the mathematical formula recognition system. The inputs to the system are symbols together with related bounding box attributes. The outputs are a LaTeX representation of the mathematical expression and the operator tree structure. To process the input of a series of recognized symbols, DRACULAE is packaged with a third-party symbol recognizer and a user interface, both of which are parts of the Freehand Formula Entry System (FFES) [19]. One motivation of our research is to generate the symbol layout class information to allow the DRACULAE to construct the spatial layout model before performing optical character recognition.

DRACULAE consists of three steps: layout pass, lexical pass and expression analysis pass. The first one constructs a baseline structure tree (BST) that represents the two-dimensional arrangement of the symbols of the mathematical expression. The lexical pass then generates a Lexed BST from the initial BST, which recognizes groups of adjacent input symbols and outputs a LaTeX representation of the input expression. Finally, the expression analysis pass produces an operator tree that describes the order and scope of operations in the input expression.

Symbol layout classes are utilized in the layout pass to construct a baseline structure tree from the input. Different layout classes have different surrounding locations. An example of BST construction for the given expression is shown in Figure 2.6. All input symbols are specified by their layout classes and bounding box coordinates. Symbol layouts are defined as the spatial relationship between the symbols within the expression, such as below, above, superscript, subscript and horizontal adjacency. The baseline is defined as a linear horizontal alignment of the symbols within the expression. Seven symbol layout classes are predefined for all mathematical symbols, which are Ascender, Descender, Center, Open Bracket, Non-Scripted, Variable Range and Root, as shown in Figure 1.1. Different layout classes have different surrounding regions where the nested baseline are located.



(a)

(b)

Figure 2.6: Baseline tree construction of the expression $(a + b^2 - p)^3 > 5$. (a) Original expression, (b) Baseline structure tree.

Compound symbols and structural symbols are recognized in the lexical analysis pass. Compound symbols refer to mathematical symbols which comprises of sequence of single input symbols. For example, *cos* consists of *c*, *o* and *s*. Structure symbols are defined as the symbols whose meaning depends on the spatial relationship between multiple baselines, such as fractions, limits, and roots.

In the expression analysis, a mathematical expression grammar and a set of tree transformations are used to create the operator tree. The grammar makes use of a modification of the traditional context-free expression grammar. A series of tree transformations are performed to determine the implicit operators and reordered operands. The form of the output of the expression analysis may be translated and executed by computer algebra system such as Maple [8].

DRACULAE was evaluated by assessing the performance of the Lexed BST recognition on the UW-III database. Two metrics were used, which are (1) the number of correctly recognized baselines, and (2) the percentage of symbols or tokens that were correctly located in their baselines. The results showed that 71% to 79% of the baselines were correctly recognized and 86% to 90% of the tokens were correctly placed by DRACULAE. In addition, DRACULAE's recognition ability was also informally tested by using the Free Formula Entry System (FFES). The results show that most users (24 of 27) found the bitmap output produced by FFES/DRACULAE from the user entered expressions to be useful.

2.3 Shape Context

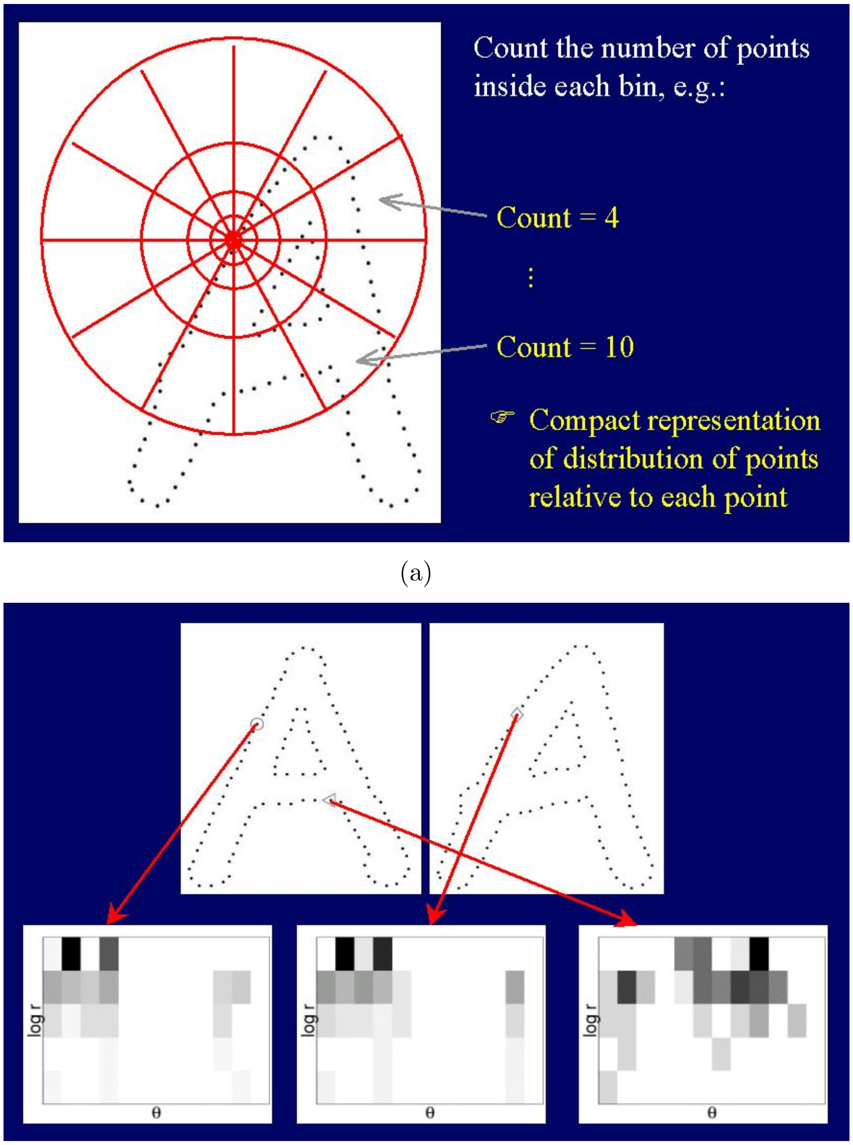
Shape context is a feature measuring the shape similarity between two objects by describing the spatial relationships between the sampled points on the shape contour of the objects [20]. The new feature developed in our work, layout context, is an extension of the shape context. Shapes are represented by a number of points uniformly sampled from the shape contour. The shape context describes a coarse spatial distribution of the rest of these points relative to a given point on the shape contour. Suppose there are k points sampled from the contour of a shape. For each point p_i on the shape, the coarse histogram labeled as h_i of the coordinates of the remaining $k-1$ sample points relative to p_i is defined as the shape context of p_i. The bins for this histogram are uniform in a log-polar space as shown in Figure 2.7(a). Based on the definition of shape context, measuring similarity between two shapes is equivalent to finding a sample point on the other shape that has the most similar shape context for each sample point on one shape. An example of the shape context of two similar points from two different shapes is shown in Figure 2.7(b).

Since shape contexts are distributions represented by histograms over a log-polar space, a cost C_{ij} for matching two points according to their shape context is calculated:

$$C_{ij} = \frac{1}{2}\sum_{i=1}^{K}\frac{[h(p_i)-h(q_i)]^2}{h(p_i)+h(q_i)} \tag{2.3}$$

The points matching cost may also include another factor based on the local appearance similarity between the two points p_i and q_i, especially for the shapes of the grey level images.

A transformation map generated by regularized thin-plate splines (TPS) is applied to map arbitrary points from one shape to the other. The thin plate spline is widely used for modeling coordinate transformation. A displacement field that maps any position in the first shape image to its interpolated location in the second shape image is generated by using two separate TPS functions [21].



Count the number of points inside each bin, e.g.:

Count = 4

⋮

Count = 10

☞ Compact representation of distribution of points relative to each point

(a)

(b)

Figure 2.7: Shape context computation and corresponding points matching [5], (a) Diagram of log-polar histogram bins used in shape context computation with 5 distance bins and 12 angle bins (b) Example shape context for reference sample points marked by ○, ◇ and ◁. Note that the shape context for ○ and ◇ are similar to each other while the shape context for ◁ is much different.

Finally shape context was tested on the digit recognition by using the K Nearest Neighbors and proved very effective with an error rate of 0.63% using the MINST database with a training set of 20,000 samples when $K = 3$.

2.4 Summary

Symbol recognition and structural analysis usually are the two main processes in a mathematical formula recognition systems. For structural analysis, layout model constructions and structure semantics analysis are the key processes to determine the expression structure and their intended semantic meaning. A number of works have examined these two issues in the past thirty years and have generated many important layout models and semantics analysis methods. DRACULAE is an implementation of an expression parser based on the layout model describing the spatial relationship between the symbols via a baseline tree structure. The input symbols of the system are preassigned a layout class that has a unique arrangement of surrounding regions. Our research focuses on the implementation of this classification. In addition, a shape-matching feature named shape context is introduced. This feature describes a coarse distribution of the points on a shape relative to a given point. Then the problem of finding two similar shapes is equivalent to finding two shapes with the smallest total matching error between the shape context for each of the sample points on the two shapes. This shape descriptor has been proven to be very robust and effective in the area of digit recognition and 3-D object recognition such as trademarks and sihouettes.

Chapter 3

Methodology

In this chapter, we present the methodology for our experiments. The classification process in our experiment includes two main steps: feature extraction and class labeling. Section 3.1 introduces the seven layout classes and their associated attributes. Section 3.2 defines the new feature *layout context* and details the feature extraction method. Section 3.3 gives a description of the Nearest Neighbor (NN) rule applied in the labeling process and the feature distance metric. Section 3.4 describes the evaluation process for the classification method. Section 3.5 lists the experiment setups and the hypotheses which these experiments are designed to test.

3.1 Symbol Layout Classes

Seven layout classes [1] are defined to describe the expected locations of neighboring symbols, which include Ascender, Descender, Center, Open Bracket, Non-scripted, Variable Range and Root, as shown in Figure 1.1.

These classes represent different centroid locations and surrounding regions associated with a symbol. The centroid of a symbol reflects the character typographic center and is used to test whether a symbol lies within a region [4]. The x center of the centroid is calculated as $\frac{X_{min}+X_{max}}{2}$, where X_{min} and X_{max} are the minimum and maximum x coordinates of the symbol bounding box. As shown in Table 3.1, the y center of the centroid is calculated according to the centroid ratio c and the symbol layout class. Surrounding regions of a symbol include: below, above, subscript and subscript. Different classes of symbols have different spatial relationships with these surrounding regions. Combinations of the attributes of each layout class are shown in Table 3.1.

Table 3.1: Symbols classes with associated centroid position and surrounding regions spatial locations based on Zanibbi,et al's layout classes [1]. H is the bounding box height ($Y_{max} - Y_{min}$) and the parameter c (centroid ratio) is used to describe the location of centroid y center in terms of the height H of the whole symbol bounding box. t is the threshold ratio which is used to describe the y coordinates of the surrounding region bottom. The centroid ratio, c, and the threshold ratio, t, are both in range [0, 0.5], with $t < c$

Symbol Class	Y-center	Threshold			
		Below	Above	SUBSC	SUPER
Ascender	cH	tH	$H - (tH)$	tH	$H - (tH)$
Descender	$H - cH$	$\frac{1}{2}H + t\frac{1}{2}H$	$H - t\frac{1}{2}H$	$\frac{1}{2}H + t\frac{1}{2}H$	$H - t\frac{1}{2}H$
Center	$\frac{1}{2}H$	tH	$H - (tH)$	tH	$H - (tH)$
Open Bracket	cH	$minH$	$maxH$	$\backslash$	$\backslash$
Non-Scripted	$\frac{1}{2}H$	$\frac{1}{2}H$	$\frac{1}{2}H$	$\backslash$	$\backslash$
Variable Range	$\frac{1}{2}H$	tH	$H - (tH)$	tH	$H - (tH)$
Root	cH	$minH$	$maxH$	tH	$H - (tH)$

3.2 Layout Context Extraction

A new feature, named *layout context*, is developed in this paper to depict the spatial relationship between a reference symbol and its local neighboring symbols within the math expression.

3.2.1 Key Points Model Representation

Since layout context describes the spatial distribution of symbol locations, the first step is to represent symbols. A number of key points from different locations of the symbol bounding box, which include sides, diagonals and center lines, are used as the model to represent an individual symbol. The key points are sampled with uniform interval by bisecting line segments on the sides, diagonals and center lines of the bounding box, as shown in Figure 3.1 to Figure 3.3. The reason for selecting a symbol bounding box is that we believe that the symbol bounding box is a simple way to describe layout. There are three typical types of key points models:

- *Model with only side points.*

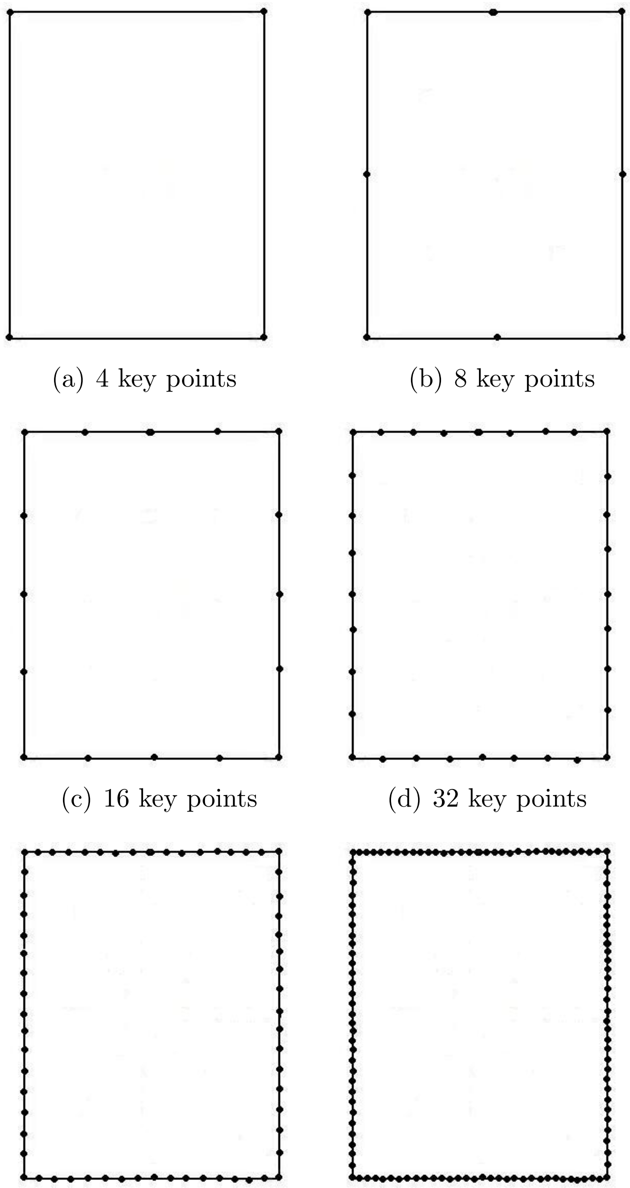
  This type of key points model includes only the points sampled from the sides of the bounding box (Figure 3.1).

- *Model with only inner points.*

  This type of key points model includes only the points sampled from the center, diagonals, and centerlines of the bounding box, which are all inside the bounding box (Figure 3.2).

- *Model with both inner and side points.*

  This type of key points model includes points sampled from both sides and the interior parts of the bounding box (Figure 3.3).



(a) 4 key points

(b) 8 key points

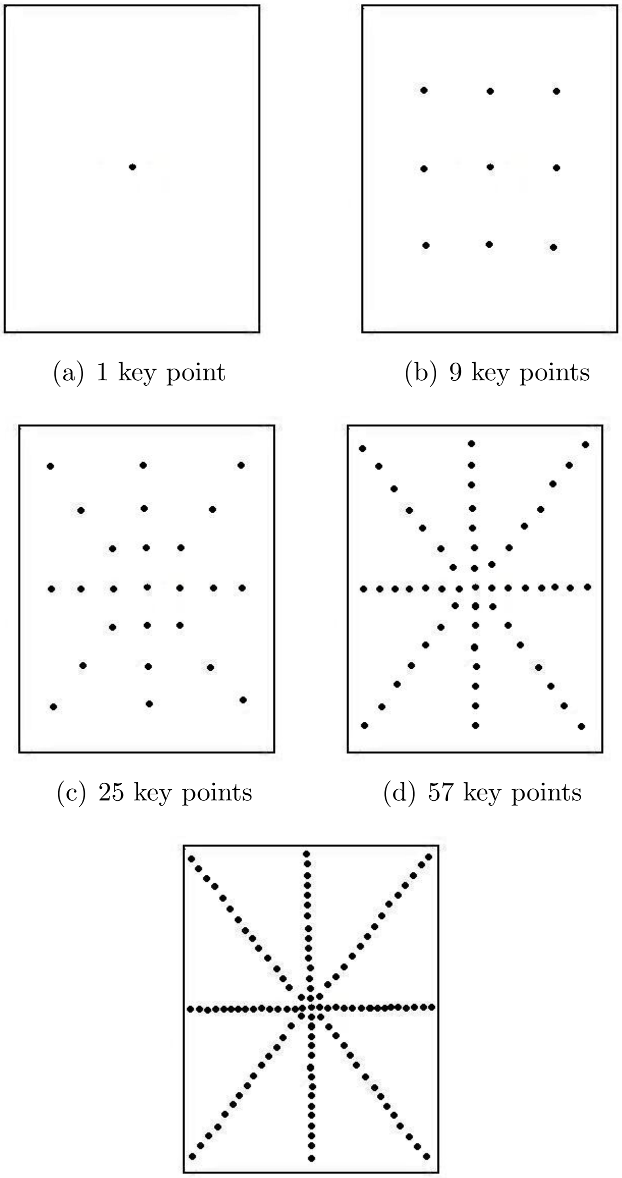
(c) 16 key points

(d) 32 key points

(e) 64 key points

(f) 128 key points

Figure 3.1: Instances of the side key points model.



(a) 1 key point

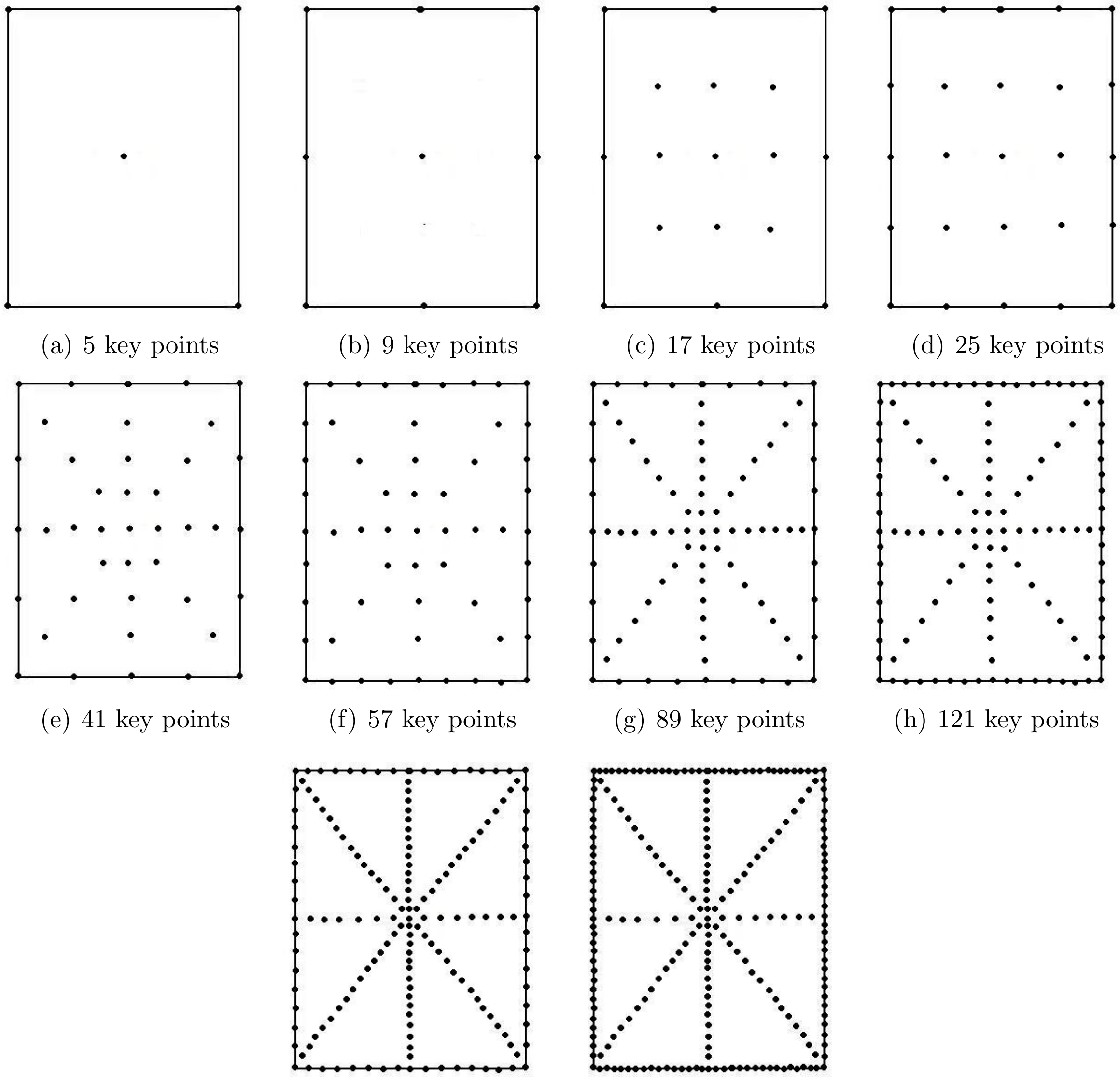
(b) 9 key points

(c) 25 key points

(d) 57 key points

(e) 121 key points

Figure 3.2: Instances of the inner key points model.



(a) 5 key points

(b) 9 key points

(c) 17 key points

(d) 25 key points

(e) 41 key points

(f) 57 key points

(g) 89 key points

(h) 121 key points

(i) 185 key points

(j) 249 key points

Figure 3.3: Instances of both side and inner key points model.

3.2.2 Neighborhood Area

After representing a symbol by a key points model, the next step is to identify the neighborhood area of the reference symbol within its math expression. The neighborhood area is identified by a circle centered at the reference symbol bounding box center. Any symbols within this area are considered to be neighboring symbols in the layout context extraction of the reference symbol.

We use r to denote the ratio between the circle radius and the unit length, which is half of the length of the reference symbol bounding box diagonal (Figure 3.4). The radius of the circle area may be changed to cover symbols in various distant areas from the reference symbol bounding box center in the later experiment. The larger the radius, the larger the circle area, which means that more symbols are covered by the circular area (relative to the reference symbol) in the later feature calculation. Several examples (Figure 3.5) present different sizes neighborhood areas of a reference symbol within a math expression.

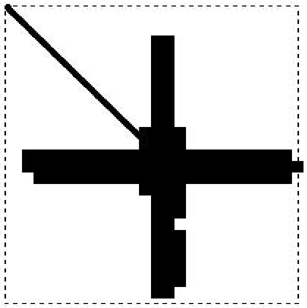
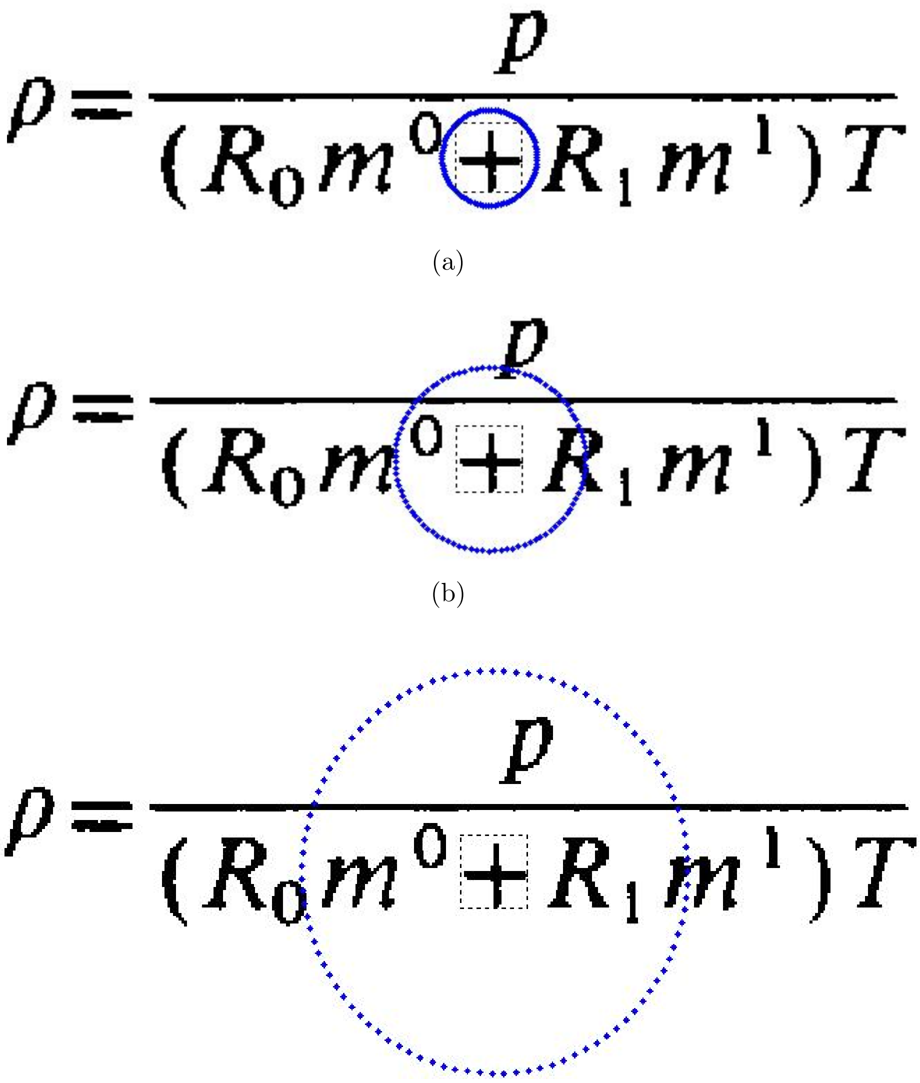


Figure 3.4: Unit length of symbol "+".



(a)

(b)

(c)

Figure 3.5: Different sizes of symbol neighborhood area ($r = 1, 2, 4$) used in layout context extraction. The parameter r denotes the ratio between the radius of the circle and the unit length: (a) The radius of the circle is equal to the unit length when $r = 1$. (b) The radius of the circle is twice of the unit length when $r = 2$. (c) The radius of the circle is four times of the unit length when $r = 4$.

3.2.3 Histogram Calculation in Log-Polar Space

After identifying the neighborhood area and representing the symbol, the layout context may be extracted by calculating the distribution of key points of the symbols within the neighborhood area relative to the reference symbol bounding box center. An example of calculating the layout context of a reference symbol "+" within a formula is shown in Figure 3.6. The key points model used in the computation can be seen in Figure 3.3(a) and the radius ratio $r = 4$. The steps for the feature extraction are stated below:

1. Compute the vectors connecting the neighboring symbol key points to the symbol center o (Figure 3.6(b)), and calculate the length and angle of each vector.

2. Divide the neighborhood region into 60 bins, consisting of 12 equal angle bins and 5 distance bins. The ratio of the radii of the five distance bins moving out from the center of the symbol are $\frac{1}{16} : \frac{1}{8} : \frac{1}{4} : \frac{1}{2} : 1$, with the whole (1) being the radius of the outermost circle neighborhood area (Figure 3.6(c)). This division of the distance bins is consistent to that of Belongie, et.al's work.

3. Compute the histogram of key points by their distance and angle relative to the reference symbol center over the 60 bins (Figure 3.6(d)). Normalize the histogram by dividing each bin by the number of key points. The resulting histogram is the layout context of the reference symbol.

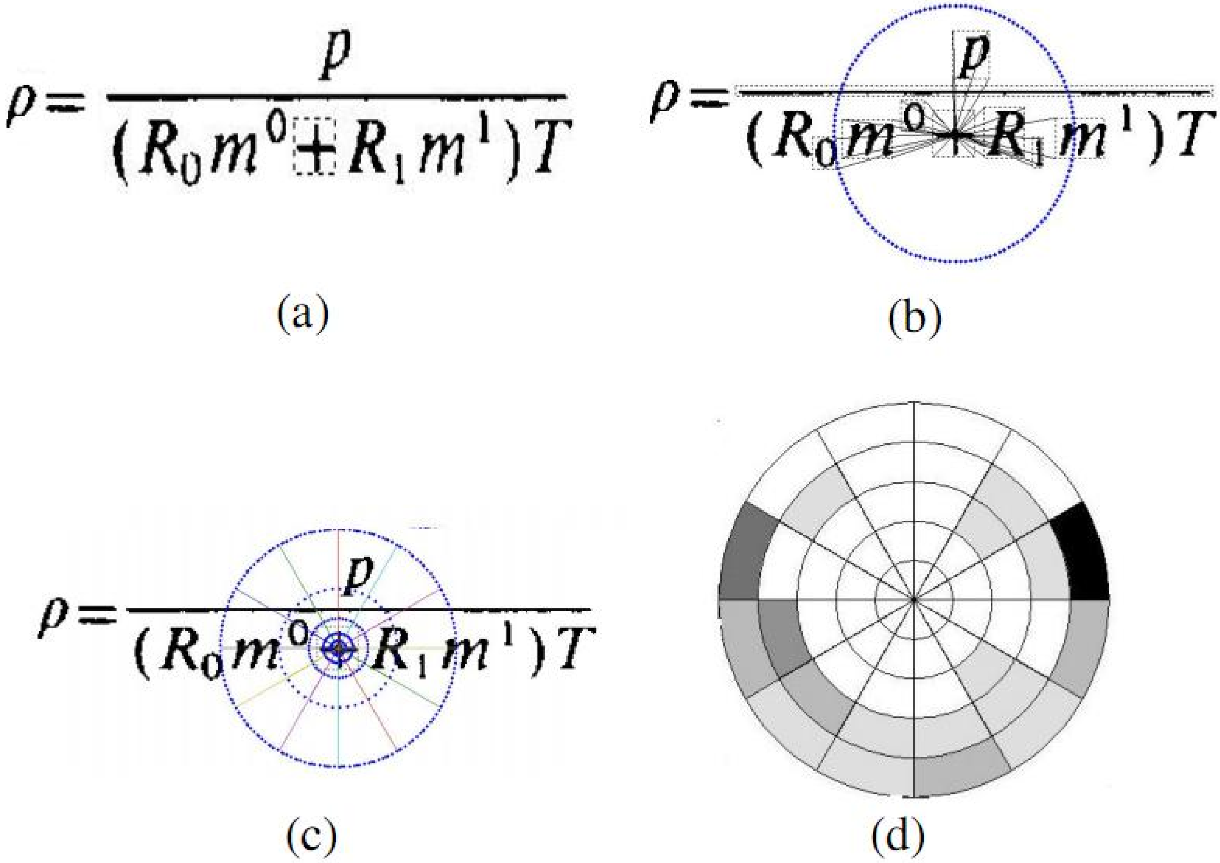


Figure 3.6: The process of calculating the layout context of the symbol “+”. (a) Expression where the reference symbol “+” lies. (b) Vectors that connect other key points within the neighborhood to the reference symbol bounding box center. (c) 60 bins of the circular neighborhood area. (d) Visual representation of the layout context of symbol “+”. The darker the bin, the larger the number of points within it.

3.3 Classification Using Nearest Neighbor Rule

After feature extraction, the class are labeled using Nearest Neighbor rule. A histogram matching cost is used as the feature vector distance metric for the nearest neighbor classification.

3.3.1 Nearest Neighbor Rule

Nearest Neighbor (NN) is a supervised classification algorithm in which the class of a new instance is that of the majority of the nearest neighbor training instances. The classifier does not depend on any probability density model but only on the labels of the nearest neighbor samples. For example, if the nearest neighbor of the reference symbol is of class Ascender, then the reference is labeled Ascender as its layout class. If the nearest neighbor symbols that belong to two different layout classes have the same feature distance to the reference, then we break the tie by randomly assigning the reference symbol one of these different layout classes of the nearest neighbor symbols.

3.3.2 Distance Metric for Layout Context Feature

The histogram matching cost between the query histogram feature and the training instance histogram feature is computed as the distance metric in the Nearest Neighbor rule. Consider two symbols p_i and q_i in an expression and let C_{ij} represent the matching cost between the layout contexts $h(p_i)$ and $h(q_i)$ of these two symbols. As the layout context is represented by histograms, it is natural to use the χ^2 metric [21] with Yates' correction [22] to represent the matching cost of the two histograms. In our work, the matching cost between the distribution of the relative coordinates for the test symbol and a distribution of its neighborhood symbol is computed:

$$C_{ij} = \frac{1}{2} \sum_{i=1}^{K} \frac{[h(p_i) - h(q_i)]^2}{h(p_i) + h(q_i)} \tag{3.1}$$

3.4 Performance Evaluation: Leave-One-Out

We adopt a classification model evaluation method named "leave-one-out" (LOO), which is an extreme situation of the k-fold cross validation. k-fold cross validation divides the data set into k subsets, each of which has an approximately equal number of data points. One of the k subsets is used as the testing set, and the remaining k-1 subsets are used as the training sets in the cross validation process. This validation process repeats k times in which each of the k subsets is used as the testing data set. In the LOO method, the number of data points of each subset becomes 1 and the k becomes the total number of data points in the sample data set. The advantage of this method is that all observations are used for both training and validation. The shortcoming is that this method may be computationally expensive if the data set is too large.

3.5 Experiment Data Set and Setup

Experiments are performed to test whether the symbol layout classification accuracy may be improved by using a large number of key points from both inside and on the boundary of the symbol bounding box and a small neighborhood size.

3.5.1 Hypotheses

Four specific hypotheses are proposed to be test in our experiments:

1. Only the symbols that are closest surrounding the reference symbol need to be included for the layout context calculation.
2. The key points model should include a large number of key points sampled from both sides and the interior of the bounding box in order to accurately represent the symbol location in the layout context extraction.
3. Including key points from both the reference symbol and the neighboring symbols gives the best classification performance.
4. For the key points model, the inner points should be sampled from some geometrical important locations of the bounding box, such as the diagonals and center lines, to represent the symbol spatial characteristic simply and effectively. Sampling the inner points in a naive way, such as uniformly across the bounding box, may cause the layout context feature less effectively in representing the spatial arrangement of the neighboring symbols relative to the reference one.

We first test hypothesis 1 by varying the radius ratio r and the number of key points p. The key points model in the experiment includes both inner and side points in the layout context computation. To test hypothesis 2, we repeat the same process of hypothesis 1 but use key points models separately with inner points alone and side points alone. Then we compare the highest classification rates of these two type of models with those of the key points model with both side and inner points. Furthermore, experiments for the key points models using only reference symbol key points and only neighboring symbols key points are performed. The classification results are also compared to that of the key points model by using both reference symbol points and neighboring symbols key points to determine if hypothesis 3 is correct. Finally, to test the validity of hypothesis 4, a grid key point model in which the inner key points are arranged in a grid pattern is used in our last experiment and the classification result is compared to that of the "cross" model, which has points on the diagonals and center lines only. Through testing these hypotheses, we can find out the how the parameters of the layout context feature affect the layout class classification rate.

3.5.2 Experiment Data Set

University of Washington English/Technical Document Images Database III [23] is used to provide both the training data and the testing data in our experiment. UWIII database is publicly available and its ground truth has been established. All mathematical symbols within expressions have corresponding $\LaTeX$ representations and symbol bounding box coordinates in XFIG format. There are 1917 symbols contained by 73 math formulas is this database. All symbols are manually labeled with their actual layout class according to the definition of the seven layout classes in Table 1.1.

3.5.3 Experiment 1: Classification Experiment with Different Neighborhood Size and Key Points Locations

Experiment 1 is designed to see the trend of classification accuracy using layout contexts with different parameter settings for the feature computation. Two parameters are involved: the ratio between the outer circle radius and the unit length (denoted as r), the type of locations where the key points are sampled (denoted as T_p).

The dependent variable in our experiment is the classification accuracy, and the independent variables are r and T_p. A summary of experimental conditions with different parameters combinations of T_p and r is shown in Table 3.2. All key points models are divided into three categories according to location type T_p and there are multiple instances of key points model for each T_p with a different number p of key points. In the left most column of Table 3.2, inner points refer to the key points sampled from the interior of the bounding box, and side points refer to the key points sampled from the sides of the bounding box. Inner and side refer to the key points sampled from both the sides and the interior of the symbol bounding box.

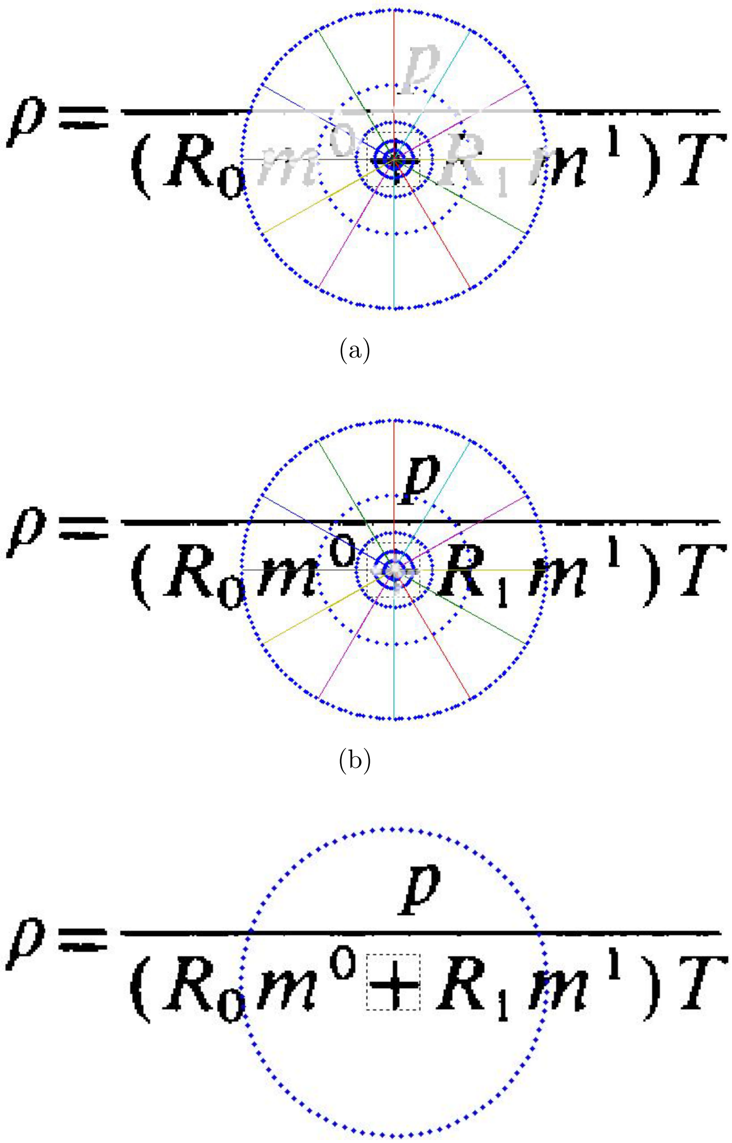
Table 3.2: A summary of different experiment conditions, where r is the ratio between the radius of the circular neighborhood and the unit length, and T_p is the type of location where the key points are sampled.

T_p	r				
	1	2	4	8	16
Inner points only	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
Side points only	Condition 6	Condition 7	Condition 8	Condition 9	Condition 10
Inner and side	Condition 11	Condition 12	Condition 13	Condition 14	Condition 15

The values of r used were 1, 2, 4, 8 and 16. For almost all the expressions in our experimental data set, the maximum value of $r = 16$ means that all other symbols within the expression would be covered by the circular neighborhood area of the reference symbol when the radius of the outer circle is 16 times the unit length (half the length of the reference bounding box diagonal). Since we have limited the neighborhood area of a symbol within the scope of its expression, there is no need to have a larger circle area (when $r > 16$). The reason that the minimum value of r is 1 is because we want the neighborhood area to include at least the reference symbol.

3.5.4 Experiment 2: Classification Experiment Using Key Points From Different Types of Symbols

To investigate the contribution of the reference symbol and the neighboring symbols to the classification performance, three different conditions are applied in our classification experiment (Figure 3.7), which are using reference symbol key points alone, using neighboring symbols key points alone and using both reference and neighboring symbols key points.



(a)

(b)

(c)

Figure 3.7: Three conditions in which the key points are from different types of symbols. (a) Only the reference symbol key points. (b) Only the neighboring symbol key points. (c) Reference symbol key points and neighboring symbol key points.

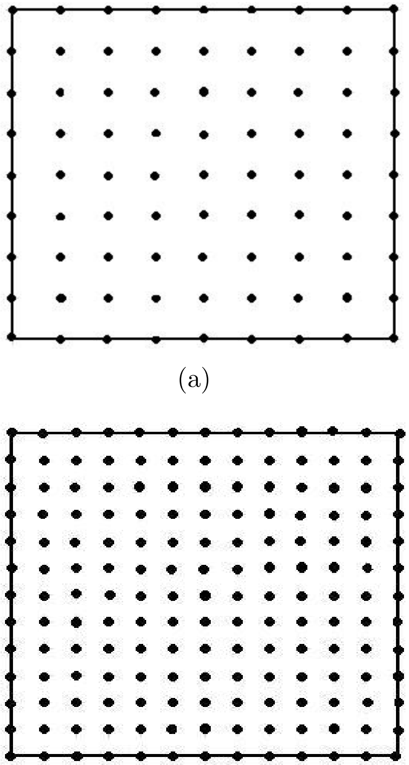
We use T_s to denote the types of symbols where the key points are taken for the layout context computation during the experiment. The parameter settings of our experiment are shown in Table 3.3. It is noted that for all the key point models in this experiment, the key points are sampled from both the sides and the interior of the symbol bounding box.

Table 3.3: A summary of different experiment conditions, where r is the ratio between the radius of the circular neighborhood and the unit length and T_s is the type of symbol where the key points are sampled.

T_s	r				
	1	2	4	8	16
Reference	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
Neighboring	Condition 6	Condition 7	Condition 8	Condition 9	Condition 10
Reference and neighboring	Condition 11	Condition 12	Condition 13	Condition 14	Condition 15

3.5.5 Experiment 3: Classification Experiment Using Grid Key Points Model

Finally, a grid key points model is designed and used in our classification experiment and the related classification results are compared to those of the "cross" key points model in which the inner points are sampled from the diagonals and center lines of the bounding box. In the grid key points model, the key points are sampled from the intersection of a $n \times n$ grid pattern of the symbol bounding box as shown in Figure 3.8.



(a)

(b)

Figure 3.8: The grid key point model (a) 8 × 8 cells intersection points. (b) 12 × 12 cells intersection points.

All conditions of the experiment are shown at Table 3.4. It is noted that the arrangement of the inner points of the grid model is quite different from the pattern of points of the “cross” model used in our previous experiments. For the “cross” model, the key points are sampled from the diagonals and centerlines of the symbol bounding box.

Table 3.4: A summary of different experiment conditions, where r is the ratio between the radius of the circular neighborhood and the unit length and T_i is the type of pattern in which the inner key points are arranged.

T_i	r				
	1	2	4	8	16
Cross	Condition 1	Condition 2	Condition 3	Condition 4	Condition 5
Grid	Condition 6	Condition 7	Condition 8	Condition 9	Condition 10

3.5.6 Summary

In this chapter, we have presented the methods for the two main layout classification processes: layout context extraction and nearest neighbor classification. In addition, we have described the experiment data set and the experimental setups for our three experiments as shown from Table 3.2 to Table 3.4. Four hypotheses are also proposed for the experiments to test. The classification rate is the dependent variable for all the experiments. The variable r is the ratio between the circular neighborhood area radius and the unit length (Figure 3.4). T_p, T_s and T_i represent different parameters in the key points model , which are the location of key points, the type of symbol where the key points are from and the arrangement pattern of the inner key points, respectively. In the next chapter, we present the result of the three experiments and discuss the effects of the parameters on the classification performance.

Chapter 4

Results and Discussion

The results for each experiment described in Chapter 3 are presented here. Section 4.1 shows the classification results with different parameter combinations of r and T_p. At the condition with the highest overall classification rate, a confusion matrix is created to describe the most confused of the layout classes. Section 4.2 presents the results of the classification experiments using key points from various types of symbols and discusses the contributions of these different types of symbols to the classification performance. Section 4.3 gives the classification result when the grid inner key point model is applied in the layout feature computation and compares it to the one where the "cross" model with inner key points on diagonals and center lines is used.

In the first experiment testing the effects of neighborhood size r and key points sampling locations T_p, the result shows that the accuracy of the layout classification is roughly 80% for a relatively small neighborhood area and key points are sampled from both inner and side of the bounding box in the layout context extraction. From the classification result in our second experiment, it is found that the reference key point and the neighboring symbols key points have different effects on the classification performance and the model using key points from both reference symbol and neighboring symbols generates the best result. In our last experiment, the results indicate that the "cross" key points model (with inner key points in the diagonals and center lines of the bounding box) may have better performance than the grid key points model. However the difference between the best classification accuracies of these two models is not big enough to be statistically significant.

4.1 Results of Experiment 1: Using Different Neighborhood Size and Key Points Locations

A series of classification experiments are performed with different radius ratio (r) ranging from 1 to 16 (1, 2, 4, 8 and 16) using three types of key point models which sample their key points from different locations: the bounding box boundary, the interior of the bounding box, and both the interior and exterior of the bounding box. A summary of classification results at all the conditions (Table 3.2) are shown in Table 4.1. For each type of key point model, there are a number of instances, each with a different number of key points, denoted as p. The classification rate for each condition presented in Table 4.1 is the highest one among all of the instances related to each condition. The tendencies of the classification rates versus r, for each type of key point model, can be seen in Appendix A.

Table 4.1: Layout classification rates at different conditions. r is that ratio between the outer circle radius and the unit length, p is the number of key points sampled to represent the symbol, T_p indicates the bounding box locations where the points are sampled.

Condition	r	p	T_p	Accuracy
Control	1	1	Inner	0.420
1	1	121	Inner	0.707
2	2	25	Inner	0.721
3	4	57	Inner	0.740
4	8	25	Inner	0.670
5	16	57	Inner	0.571
6	1	128	Side	0.689
7	2	128	Side	0.739
8	4	128	Side	0.733
9	8	16	Side	0.662
10	16	4	Side	0.559
11	1	89	Side and Inner	0.735
12	**2**	**89**	**Side and Inner**	**0.792**
13	4	89	Side and Inner	0.748
14	8	41	Side and Inner	0.662
15	16	249	Side and Inner	0.557

As shown in Table 4.1, the control condition is that the radius of the outermost circle is half of the length of the bounding box diagonal ($r = 1$, encircling the bounding box) and the key point model uses only bounding box centers ($p = 1$). As seen in Table 4.1, accuracy ranges from 42.0% (control) to 79.2% (condition 12). Performance is best when $r = 2$ and both inner and side points are used with $p = 89$.

A confusion matrix for the best condition in Table 4.1 (condition 12) is shown in Table 4.2. The last row of Table 4.2 contains the most frequent confusion for each layout class.

Table 4.2: Confusion matrix for the best condition (12) in Table 4.1

				Correct	Class			
Predict Class	Ascender	Descender	Center	Open Bracket	Non-script	Variable Range	Root	Predicted Frequency
Ascender	**82.8%**	11.8%	12.0%	5.0%	6.3%	15.9%	0.0%	660
Descender	3.2%	**69.2%**	4.2%	2.1%	2.3%	2.2%	0%	144
Center	7.3%	13.3%	**75.6%**	10.0%	9.5%	4.5%	4.5%	501
Open Bracket	0.9%	0.7%	2.8%	**82.0%**	0.7%	2.2%	0	139
Non-script	5.3%	3.9%	5.0%	0	**80.3%**	9.0%	0	413
Variable Range	0.4%	0.7%	0.4%	0.7%	0.7%	**65.9%**	0.0%	39
Root	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	**95.4%**	21
Actual Frequency	657	127	500	139	428	44	22	1917
Common Confusion	Center	Center	Ascender	Center	Center	Ascender	Center	

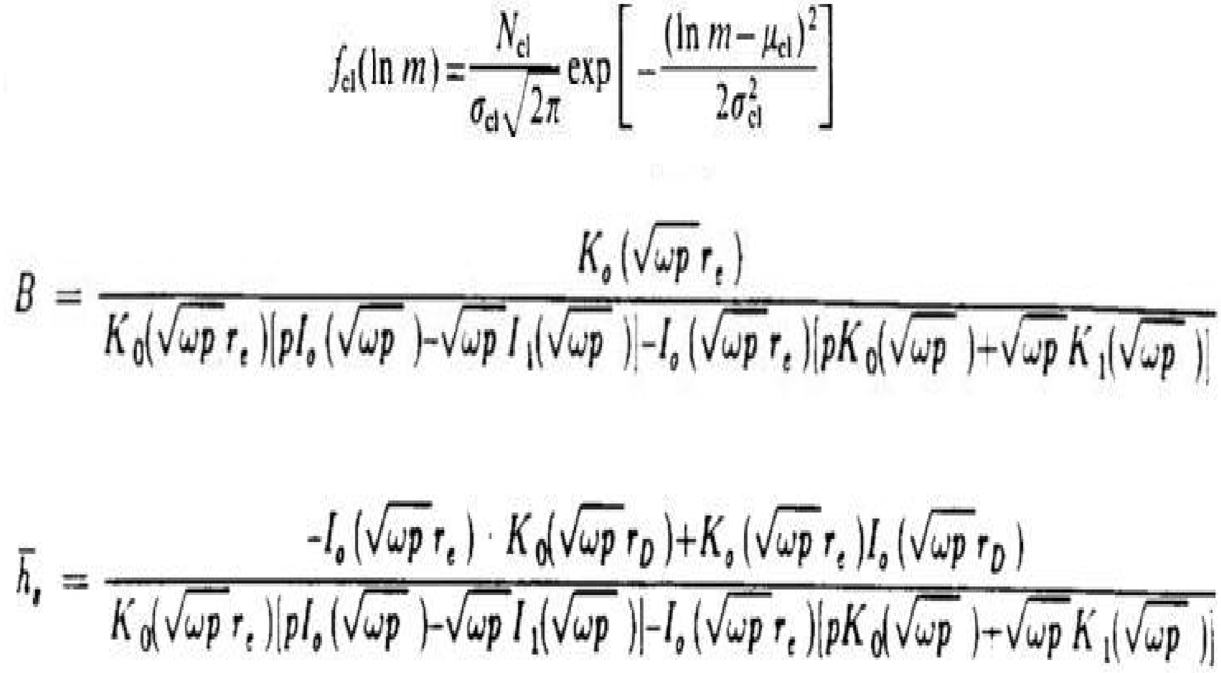
The average layout contexts of the seven symbol layout classes are shown in Figure 4.2. Some layout characteristics of the symbols are reflected from these pie chart representations.

1. The outer bins are usually darker than the inner bins, which indicates that there are more neighboring symbols distributing in more distant neighborhood area than in the near neighborhood area relative to the reference symbol. One reason is that the outer bins have larger areas because of the larger radius ratios of the five distant bins: $\frac{1}{16} : \frac{1}{8} : \frac{1}{4} : \frac{1}{2} : 1$ (Figure 3.6(c)). As a result, the more distant the bin is from the reference symbol, the larger the bin's area so that it generally contains more

neighboring symbols .

2. The bins in horizontal areas (leftmost and rightmost bins) are darker than those in the vertical areas (top and bottom bins). One reason is that the symbols in expression are more likely arranged horizontally than arranged vertically.

3. For the Root class, differences between the densities of the leftmost-upper and rightmost-upper bins are much larger than that of other layout classes; this is partly an artifact of our data set, where roots are often located in the denominator of a fraction (Figure 4.1).

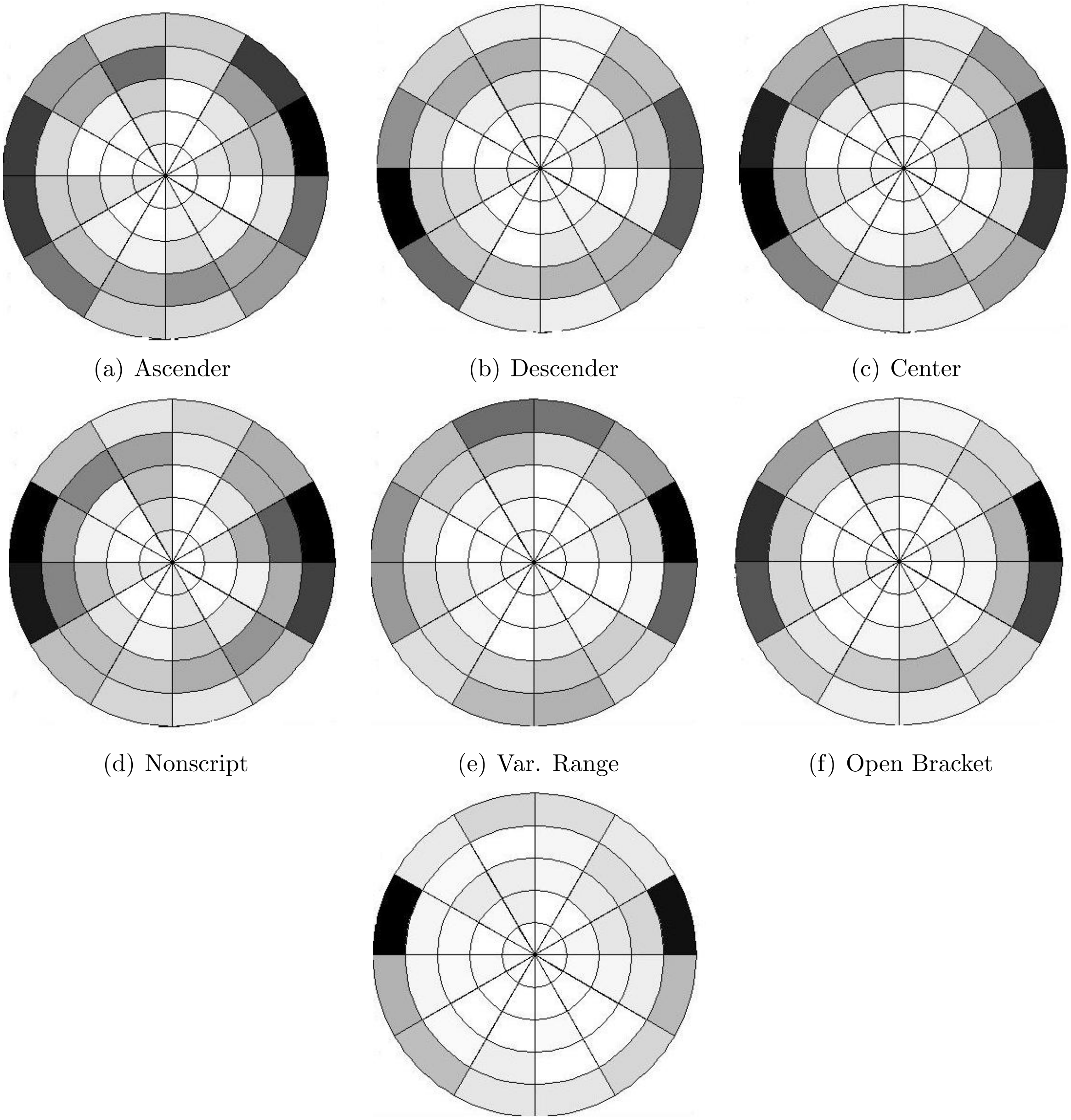


$$f_{\mathrm{cl}}(\ln m) = \frac{N_{\mathrm{cl}}}{\sigma_{\mathrm{cl}}\sqrt{2\pi}} \exp\left[-\frac{(\ln m - \mu_{\mathrm{cl}})^2}{2\sigma_{\mathrm{cl}}^2}\right]$$

$$B = \frac{K_o(\sqrt{\omega p}\, r_e)}{K_0(\sqrt{\omega p}\, r_e)[pI_o(\sqrt{\omega p}) - \sqrt{\omega p}\, I_1(\sqrt{\omega p})] - I_o(\sqrt{\omega p}\, r_e)[pK_0(\sqrt{\omega p}) + \sqrt{\omega p}\, K_1(\sqrt{\omega p})]}$$

$$\bar{h}_e = \frac{-I_o(\sqrt{\omega p}\, r_e) \cdot K_0(\sqrt{\omega p}\, r_D) + K_o(\sqrt{\omega p}\, r_e) I_o(\sqrt{\omega p}\, r_D)}{K_0(\sqrt{\omega p}\, r_e)[pI_o(\sqrt{\omega p}) - \sqrt{\omega p}\, I_1(\sqrt{\omega p})] - I_o(\sqrt{\omega p}\, r_e)[pK_0(\sqrt{\omega p}) + \sqrt{\omega p}\, K_1(\sqrt{\omega p})]}$$

Figure 4.1: Document images of expressions in the UWIII experiment data set containing square roots.



(a) Ascender

(b) Descender

(c) Center

(d) Nonscript

(e) Var. Range

(f) Open Bracket

(g) Root

Figure 4.2: Average layout context histogram for each layout class.

The accuracy of the classification for each of the seven classes is described in Figure 4.3, which is the ratio between the number of symbols that are correctly assigned to their layout class against all the symbols that actually belong to this class. The Root class has the highest accuracy, possibly because the layout contexts for root symbols are quite distinct from the other classes (Figure 4.2(g)).

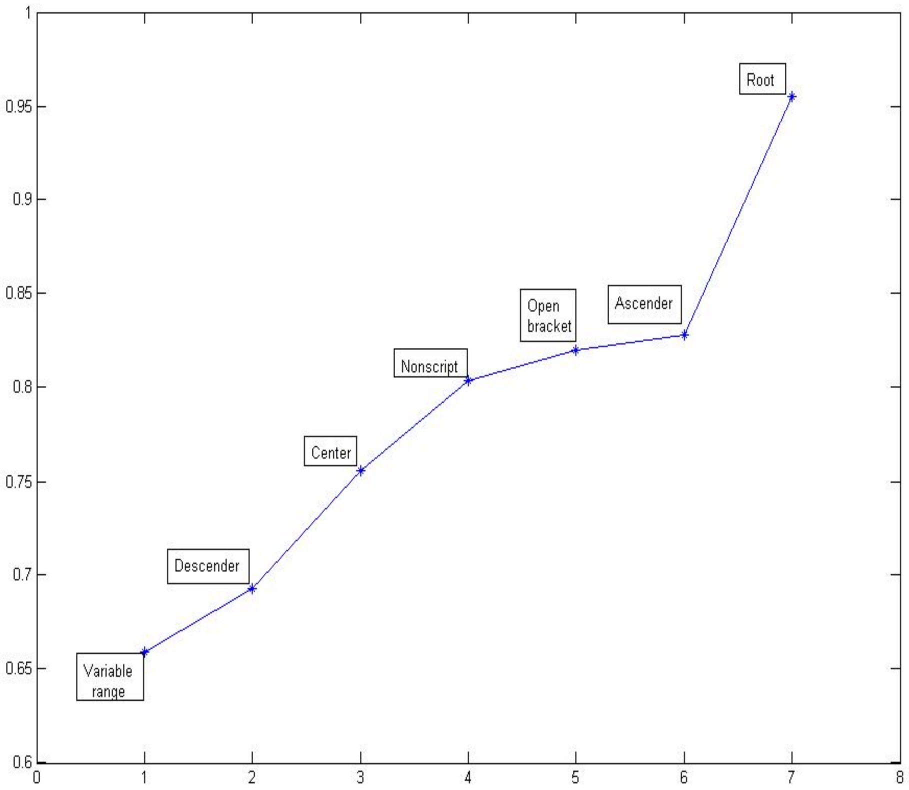


Figure 4.3: The accuracy of the classification result of the seven layout classes.

4.1.1 Discussion

From the summary in Table 4.1 and the tendency of classification rates at different r in Appendix A, it is found that including key points of neighboring symbols that are very close to the reference symbol gives the highest classification accuracy for all the three types of key points models. These key points are located within the circle whose radius is the twice the unit length of the reference symbol bounding box ($r = 2$). Accuracy decreases when larger or smaller neighborhood area is attempted. This maybe because the key points of neighboring symbols that are distant from the reference symbol may induce spurious or unhelpful information, which degrades the feature performance.

It is also observed from the confusion matrix (Table 4.1) that the Center layout class is the most frequently confused. This may be because the definition of the Center layout class (Table 3.1), includes all symbols that do not belong to the other six classes. This may include some symbols that we had not have anticipated (e.g.,$\hat{\gamma}$ may be confused with Descender class symbol γ). These unexpected symbols may have similar layout context to that of other layout class symbols and thus make some of the other six layout symbols classified as Center. Wrongly assigning a symbol a layout class may lead to errors in judging the spatial relationship between the reference symbol and its neighboring ones. For example, in the step of the baseline structure tree construction of DRACULAE, the Descender class symbol usually has a higher threshold of the subscript region than the Center class does (Figure 1.1 and Table 3.1). As a result, if a symbol of the Descender class is mistakenly classified as the Center class, the neighboring symbol that lies in the subscript region of the reference one may not be labeled as the subscript but the horizontal one of the reference symbol in the generated baseline tree.

As it is shown in Figure 4.3 that the Root class has the best classification accuracy. One explanation is that the neighborhood area for square roots is usually larger than that of other layout classes for a given r due to a longer bounding box diagonal. Consequently, there are more symbols covered in the ring zone between the two most distant circles from the bounding box center.

A comparison of the highest classification rate of the three types of key points models at each radius r is shown in Figure 4.4. The parameters setting for the best classification rate for different radius ratio r can be seen in Table 4.1. The results show that including points from both the boundary and interior of the bounding box in the layout context extraction usually generates the smallest classification error compared to including key points from either side or interior of the bounding box when the neighborhood area is not too big ($r < 8$) to include symbols that are too far away from the reference. In addition, the number of key points (parameter p) used by a specific key point model instance in the layout context also affects the classification accuracy as shown in Appendix A. In general, a large number of points are needed to represent the symbol bounding box in the layout context extraction. Using too few key points ($p < 5$) would degrade the classification accuracy. This is because the layout context feature will not accurately reflect the spatial distribution of symbols if the number of key points included in the histogram is small. However the classification accuracy may also degrade when too many points are included in the key points model. One explanation is that including too many points may make the histogram values become too large for all the symbols and in turn the layout context features of the symbols belong to different classes are less distinctive. As a result, the classification accuracy decreases.

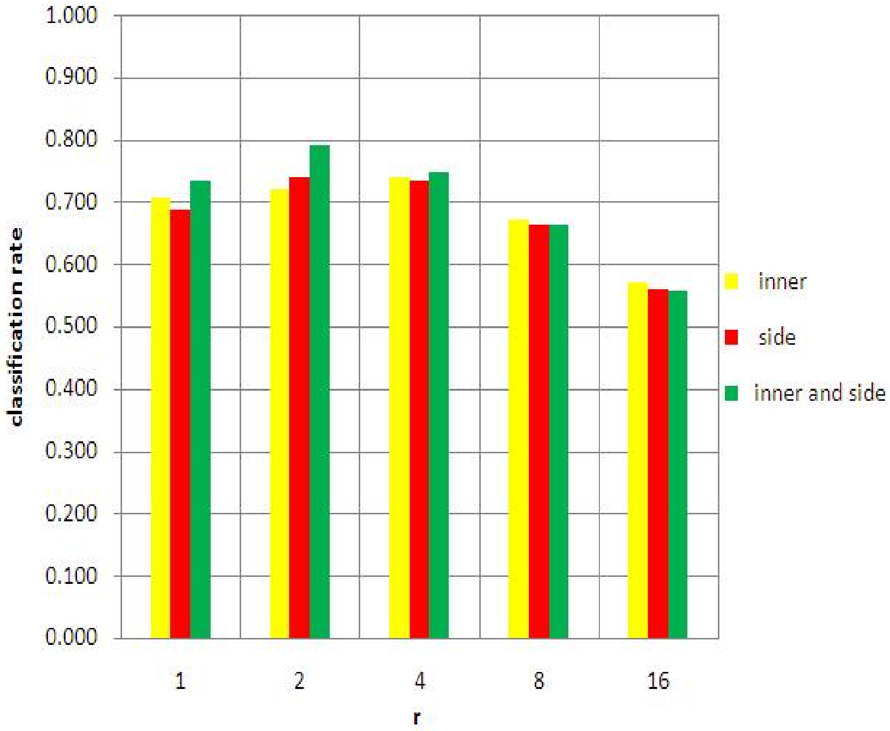


Figure 4.4: Comparison of the highest classification rates using points from three different locations of the symbol bounding box: side, interior, and both side and interior.

4.2 Results of Experiment 2: Using Key Points From Different Types of Symbols

A summary of the classification results of Experiment 2 is shown in Table 4.3. The tendencies of classification accuracies versus r by using key points from different types of symbols are shown in Appendix B. The comparison of the highest classification rates between reference symbol model, neighbor symbol model, and both reference and neighbor symbol model, at different r, is shown in Figure 4.5. It is noted that both inner and side key points are sampled in the key points model in this experiment.

Table 4.3: The classification rate versus r using three types of symbol key points: reference symbol points, neighborhood symbol key points and both reference and neighborhood symbol key points. r is the ratio between the radii of the circle neighborhood and the unit length, and p is the total number of key points applied in each instance. T_s denotes the type of symbols from which the key points are sampled.

Condition	r	p	T_s	Accuracy
1	1	89	Reference	0.625
2	2	249	Reference	0.624
3	4	249	Reference	0.621
4	8	185	Reference	0.619
5	16	249	Reference	0.615
6	1	185	Neighboring	0.570
7	2	185	Neighboring	0.727
8	4	185	Neighboring	0.717
9	8	249	Neighboring	0.648
10	16	185	Neighboring	0.557
11	1	89	Reference and Neighboring	0.735
12	**2**	**89**	**Reference and Neighboring**	**0.792**
13	4	89	Reference and Neighboring	0.748
14	8	41	Reference and Neighboring	0.663
15	16	249	Reference and Neighboring	0.557

4.2.1 Discussion

From Table 4.3, it is found that the best classification result (79.2%) appears at the condition that both reference key points and neighboring symbols key points are used when $r = 2$. This is consistent to the hypothesis 2 that using both reference symbol and neighboring symbol key points generates the best classification result.

Based on the comparison result shown in Figure 4.5, it is found that using key points from the reference symbol only, the highest classification rate is around 63%, which is much lower (nearly 17%) than the best one (79.2%). This is possibly because the extracted layout context by using reference symbol key points alone reflects only the spatial characteristic of the reference symbol itself, without any surrounding symbols

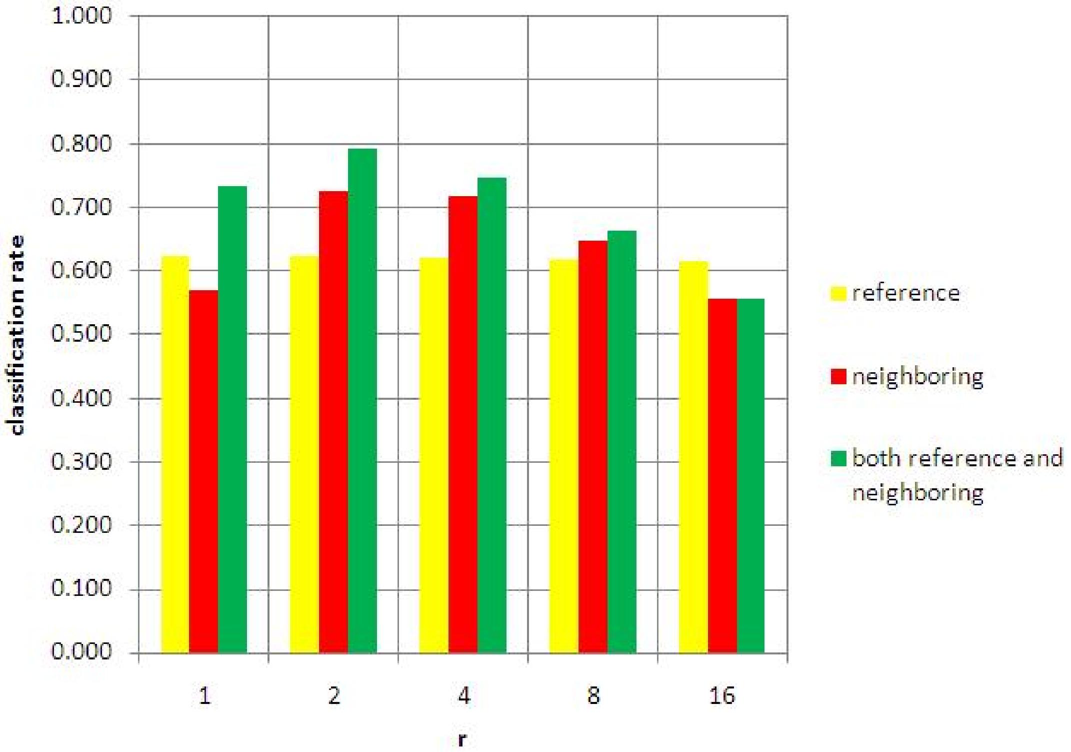


Figure 4.5: Comparison of highest classification rates for key points taken from the reference symbol alone, neighboring symbols alone, and both reference and neighbor symbols for different neighborhood area sizes at various radius ratio r.

spatial distribution information. As a result, the feature may not differentiate symbols of two different layout classes that are very similar in their individual spatial characteristic, such as aspect ratio and size, but with different surrounding symbols arrangements.

Using the neighboring symbols alone, it is found that the highest classification rate (72.7% in condition 7) is more than 10% higher than using the reference symbol only (62.5% in condition 1). This indicates that the distribution of the neighboring symbols relative to the reference symbol, which mainly reflects the layout context, plays a more important role in the process of classifying the symbols into their proper layout class than the spatial characteristic of the reference symbol itself. Furthermore, the classification rate versus the radius ratio r of this model follows the same trend that the layout classification rates increase as r changes from 1 to 2 to 4 and then decreases as r changes from 2 to 4 to 16 as shown in Appendix B. This indicates that the neighborhood symbols contribute to the variation of the classifier performance at different radii.

Additionally, it is also noticed that when the neighborhood size is very large ($r = 16$), the highest classification rate achieved by using models with points only from the neighboring symbols and models with points from both the reference symbol and the neighboring symbols are lower than those of the model with key points from the reference symbol only (Figure 4.5). One explanation for this special case is that including too many symbols that are far from the reference symbols may degrade the feature performance badly because of the irrelevant symbol key points induced in the layout context feature. For example, in the expression $(p^2 - b^2) + 4 - \sqrt{c}$, to exact the layout context of symbol p and b, there is no need to cover the symbol c since c is too far away from p and b so that there is no much difference between the spatial positions of c relative to p and b. As a result, adding key points from the symbol c has little use to make the layout context features of p (which belongs to the descender class) and b (which belongs to the ascender class) to be more distinctive from each other and may even degrade the effectiveness of the feature in the classification.

4.3 Result of Experiment 3: Using Grid Key Points Model

The summary of the classification experiment result using a grid key points model is shown in Table 4.4. The trends of classification rates versus r using the grid key points model are shown in Appendix C.

Table 4.4: The highest classification rates versus r using two types of key points models with different inner points arrangements: grid key points model and "cross" key points model (inner points are sampled on the bounding box diagonals and centerlines). r is the ratio between the radii of the circular neighborhood and the unit length and p is the number of the key points model. T_i denotes the type of arrangement pattern for the inner points of the key points models.

condition	r	p	T_i	Accuracy
1	1	89	cross	0.735
2	**2**	**89**	**cross**	**0.792**
3	4	89	cross	0.748
4	8	185	cross	0.662
5	16	185	cross	0.553
6	1	169	grid	0.704
7	2	81	grid	0.753
8	4	81	grid	0.736
9	8	81	grid	0.660
10	16	169	grid	0.551

A comparison between the highest classification rate using the grid model and the "cross" models (in which the inner points are sampled from diagonals and center lines of the bounding box) is shown in Figure 4.6. In this experiment, the key points are sampled from both the sides and interior of the symbol bounding box. In addition, the key points are from both the reference symbol and the neighboring symbols.

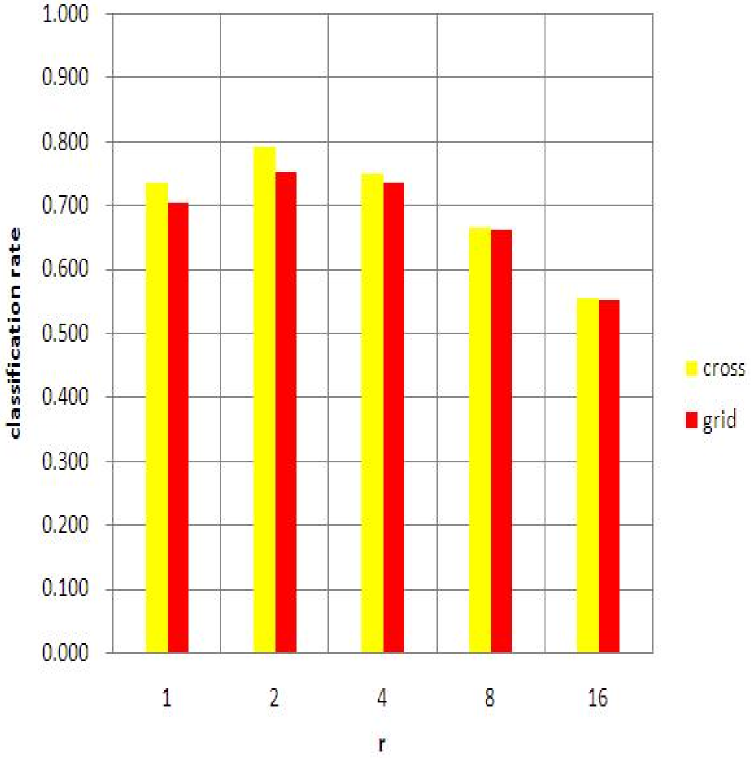


Figure 4.6: Comparison of the highest classification rates between the "cross" key points model and the grid key points model at different r.

4.3.1 Discussion

The comparison between the “cross” key points model and the grid key points model (Figure 4.6) shows that the “cross” key points model has a better performance than the grid key points model. The gap between the highest classification accuracies of these two types of models is 4% when $r = 2$ (the highest classification accuracy of the “cross” one is 79.2% (condition 2) while that of the grid key points models is 75.2% (condition 7). Note that the difference between the number of points in these two conditions (p=89 in condition 2 and $p = 81$ in condition 7) may be ignored since the histogram based feature distance computed by using the χ^2 metric in our classification would not change by increasing only a small number of key points (smaller than 10). In addition, Table 4.4 shows that the grid key points model generates the best classification rate when $p = 81$ with $r = 2$. Increasing the number of key points in the grid key points model does not improve the classification rate as shown in Figure C.1 (when $p = 169, r = 2$). It is noticed that the best classification rate of “cross” key points model is higher than that of the grid one. One possible reason is that the inner points in the grid key points model are arranged too densely across the bins, which may cause the layout context of the symbols of different classes less distinct. However, since the difference between the highest classification rates between these two types of models is less than 5%, which may not be statistically significant, other types of key points model would be used to compare to the grid key points model to find out whether sampling the inner points from some special geometrical position of the bounding box would make the layout context feature more effective in the future work.

4.4 Summary

Our experimental results shows that the classification accuracies of the layout class of the mathematical symbols could be improved to nearly 80% by covering the closest surrounding neighboring symbols in a relatively small size of circular neighborhood and sampling a fair large number of key points from the bounding box. However, including too many points (more than 89 points) in the key points model may also degrade the classification performance. The result of Experiment 1 supports hypothesis 1.

In addition, the comparison between the highest classification rates using key points models which sample the points from different bounding box locations in Experiment 1 shows that building a key points model in which the key points are sampled from both the sides and from the interior of the bounding box gives the best classification result. This result is consistent to the hypothesis 2.

We have also studied the contribution of reference symbol key points and neighboring symbols key points to the classification performance in Experiment 2. Three conditions are applied in the classification experiments: using key points from the reference symbol alone, from neighboring symbols alone, and from both reference symbol and neighboring symbol key points. The results show that including the key points from both the reference symbols and neighboring symbol gives better performance than using either neighboring symbols or reference symbols alone, which is consistent with hypothesis 3. In addition, the neighboring symbols play a more important role than the reference symbol in the symbol layout classification.

Finally, a special grid key point model is designed and tested to see the effect of the sampling inner points on the final classification performance. Although the result shows that the "cross" key point model with inner points sampled from diagonals and center lines has better performance than that of the grid, more key points models are need to be tested in the future work because the difference between the highest classification rates of these two types of key points models is fairly small and may not be statistically important to support the statement in hypothesis 4.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Thesis Statement: One can increase the accuracy of symbol layout classification using layout contexts, by using a large number of key points from both inside and on the boundary of the symbol bounding box and a small neighborhood.

We have developed a new feature named *layout context* and have used it to successfully classify most mathematical symbols within the formulas of our ground truth database. The classification into seven layout classes exhibits an overall accuracy of 79.2% by properly identifying their neighborhood area so that only the closest surrounding symbols are covered and by using a key points model consisting of points from both sides and interior of the symbol bounding box.

Furthermore, we have also conducted experiments to see the contribution of reference symbol and neighborhood symbols in isolation to the layout context feature. The comparison shows that using points from both reference symbols and neighborhood symbols provides the best classification accuracy.

Finally, we have performed experiments to compare the classification performance between the ”cross” key points model and the grid key points model. The results show that the “cross” key points model with diagonal and centerline points outperforms the one that using a grid. However, since the difference between the highest classification rates by using these two models is fairly small and may not support or reject hypothesis 4, more types of key points models are need to be tested in the classification experiments and the results are to be compared to that using a grid model.

5.2 Future Work

The layout context feature may be enhanced by adding a weighted parameter for the distance between the key point within the neighborhood area and the center of the bounding box of the reference symbol. The longer the distance, the smaller the weight added to it. This may decrease the redundant information induced by the distant key points. In addition, more types of key points models pattern can be tested by sampling the points from other locations such as the symbol contour and foreground pixels.

Other learning algorithms may be applied in the classification process. Although nearest neighbor algorithm is easy and effective, it is based merely on the distance between the training instance and query instance. More complicated but powerful algorithms such as neural networks can learn the nonlinear relationship between independent variables and dependent variables and can also apply more weight to the more effective feature element based on the intermediate feedback.

One might consider combining a global layout context, which describes all other symbols within the expression relative the reference symbol, with the local layout context to form a new feature for the layout classification. The reason is that this global layout context describes the visual characteristics of the expression, such as the length and width of the expression and the number of symbols in the expression; this might be helpful in classifying the layout of the symbol.

Some visual features, such as aspect ratio and shape context, might be employed in the classification.

At last, we may also try to find some experiment data sets that have a large number of symbols and expressions, which would make the classification process more reliable and robust if more types of mathematical symbols are involved.

Appendices

Appendix A

Classification Rates Using Different Points Locations

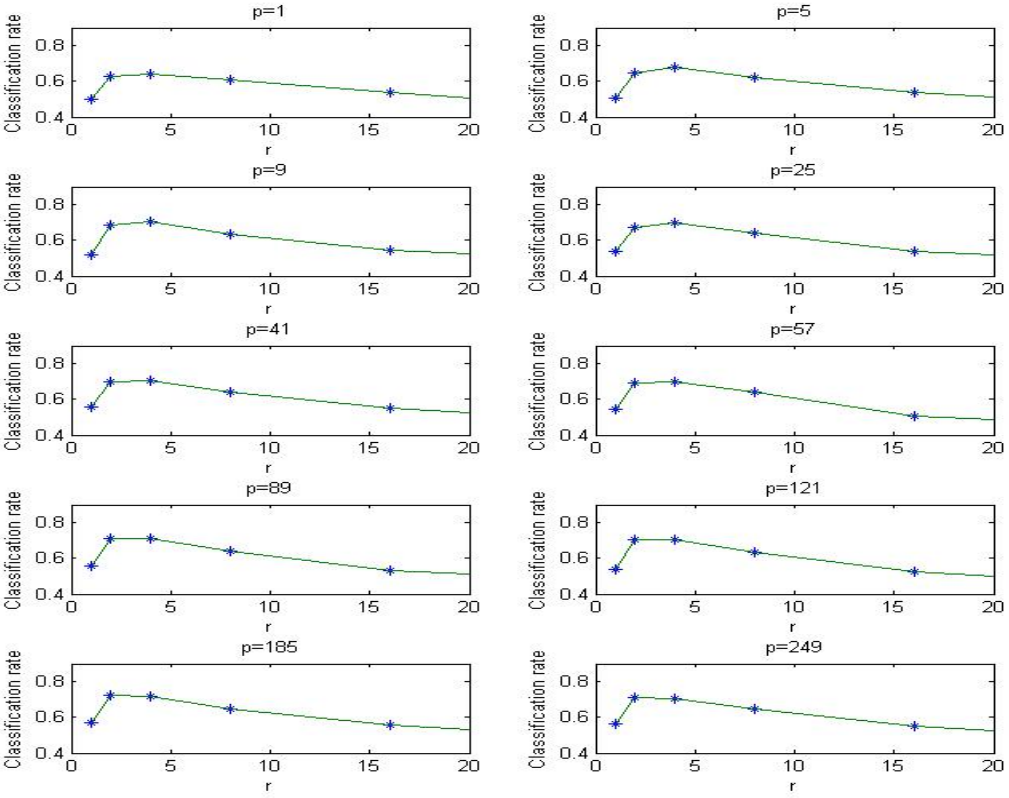


Figure A.1: The classification rates versus different r using the inner key points model.

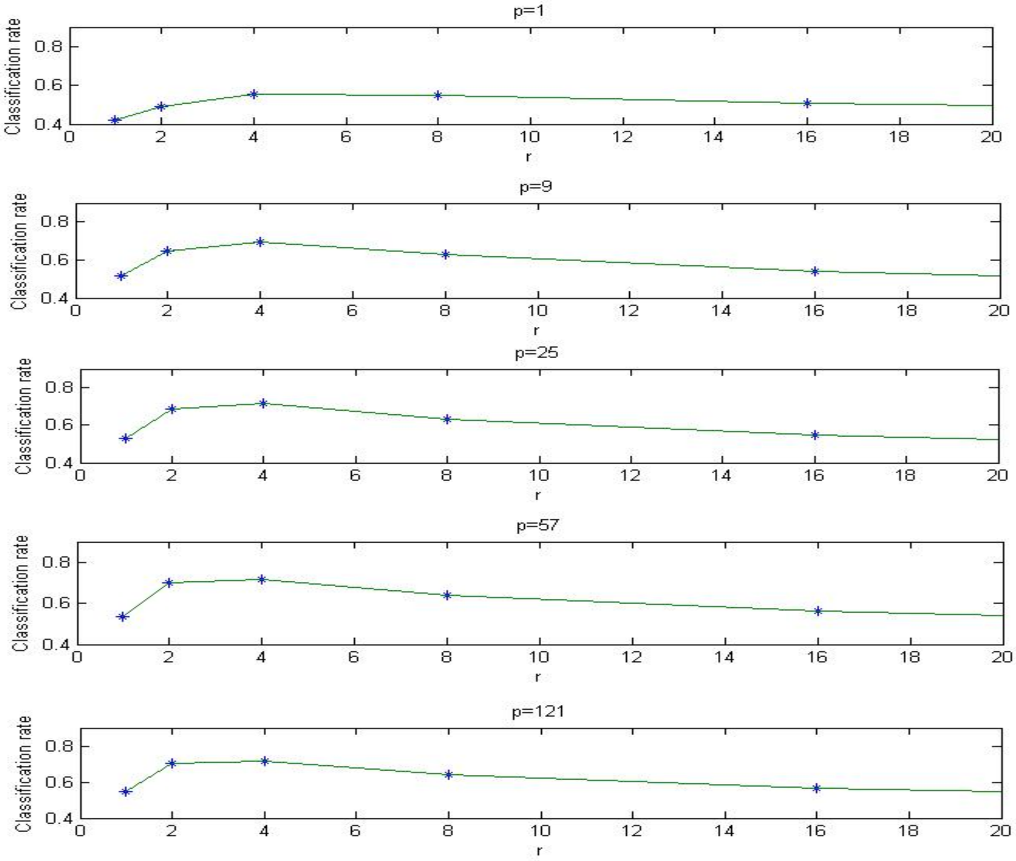


Figure A.2: The classification rates versus different r using the side key points model.

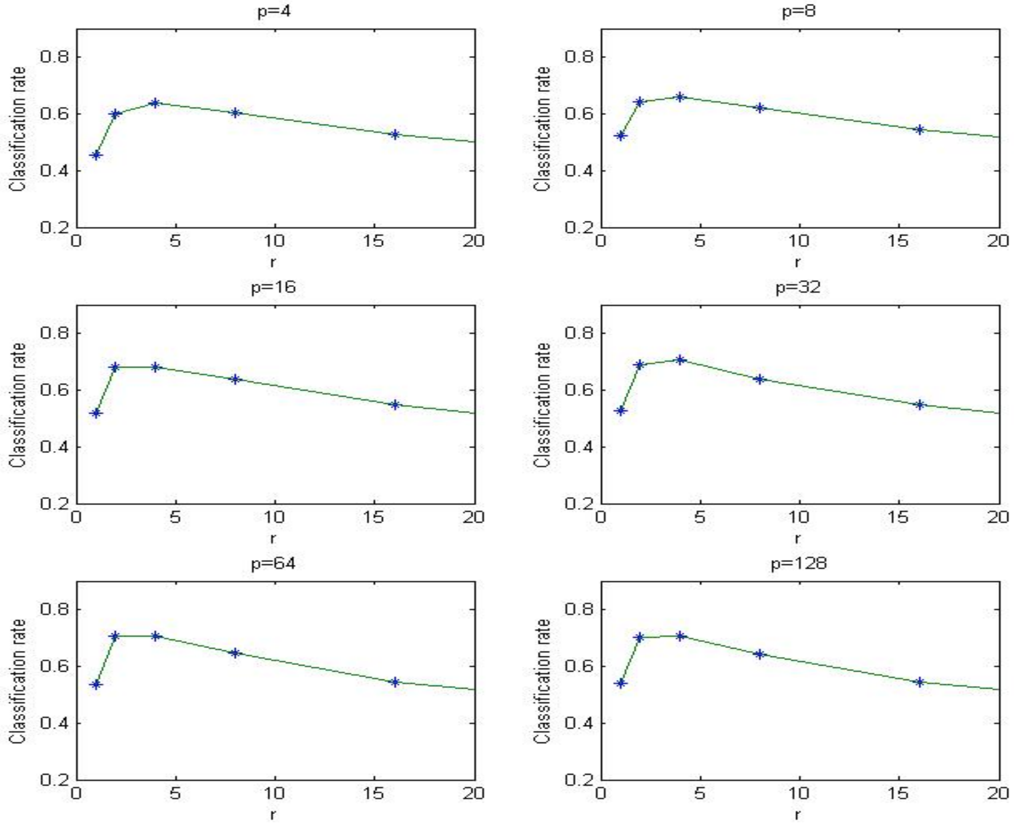


Figure A.3: The classification rates versus different r using both inner and side key points model.

Appendix B

Classification Rates Using Different Type of Symbols

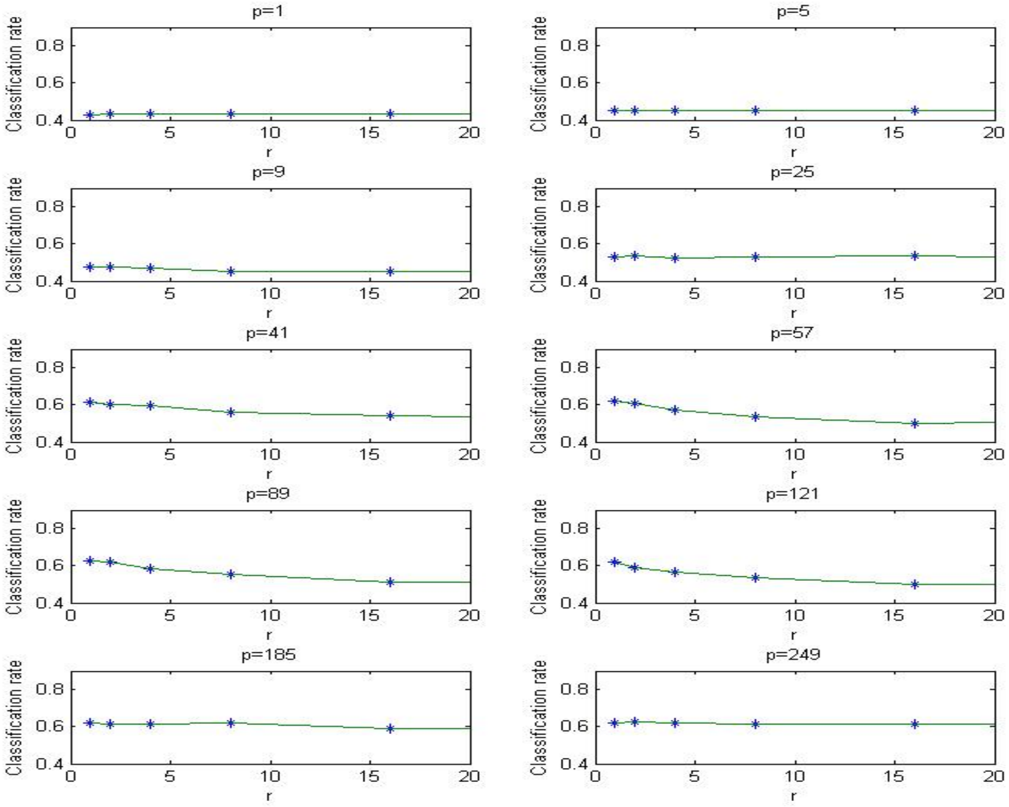


Figure B.1: The classification rates versus different r using reference symbol key points.

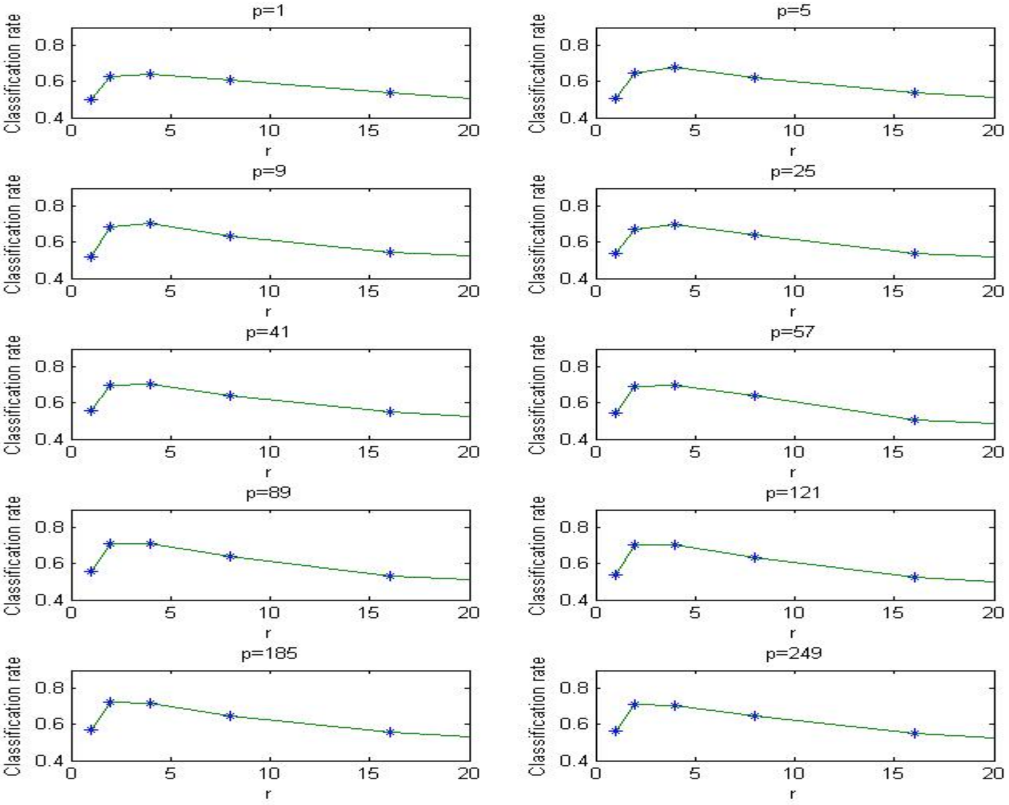


Figure B.2: The classification rates versus different r using neighboring symbol key points.

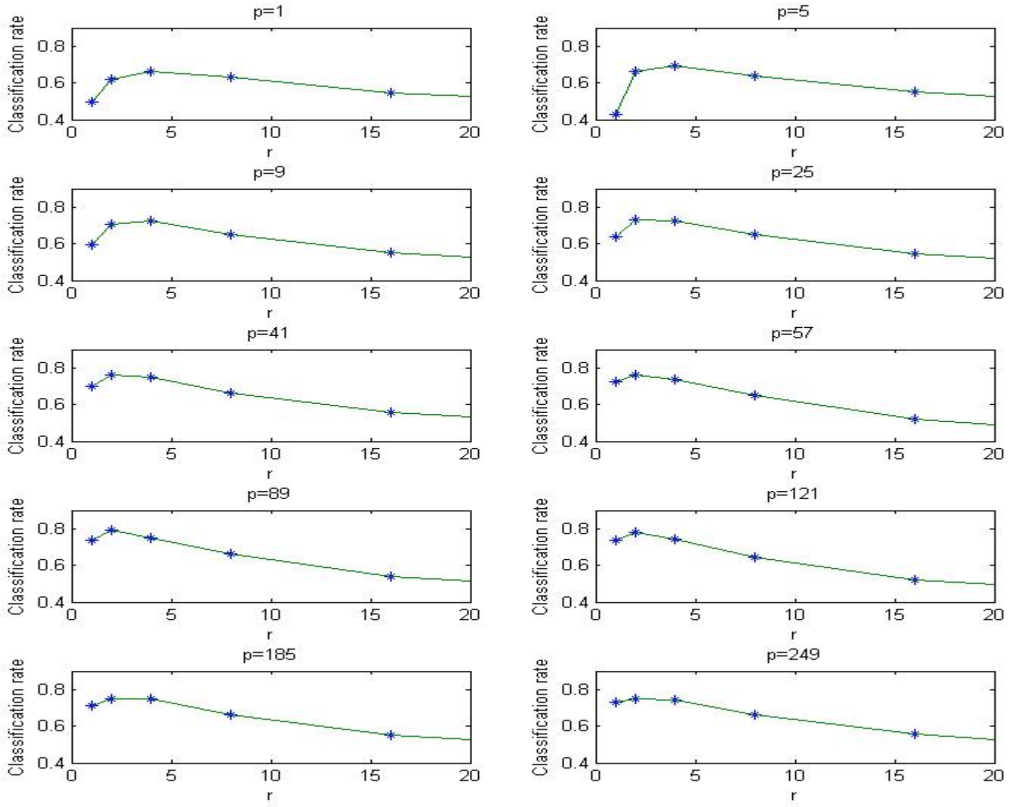


Figure B.3: The classification rates versus different r using both reference and neighboring symbols key points.

Appendix C

Classification Rates Using Grid Key Points Model

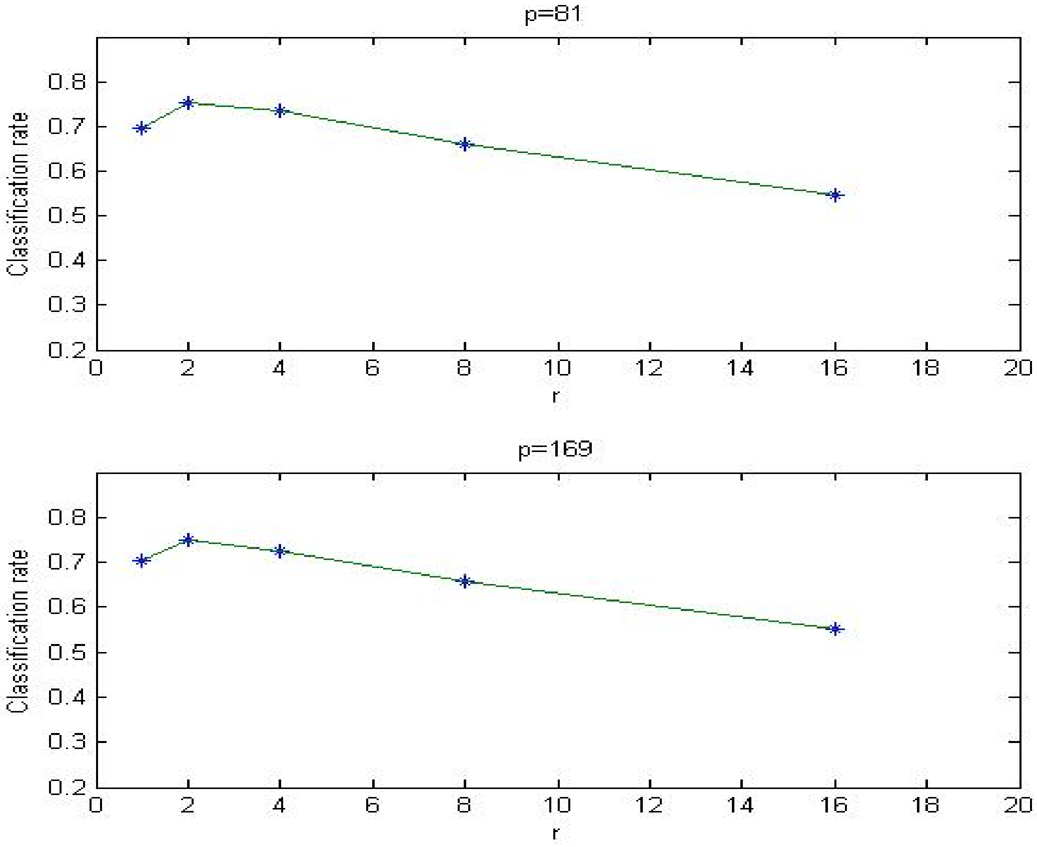


Figure C.1: The classification rates versus different r using grid key points model.

Bibliography

[1] R. Zanibbi, D. Blostein, and J.R. Cordy, "Recognizing mathematical expressions using tree transformation," *IEEE Transactions On Pattern Aanlysis and Machine Intelligence*, pp. 1455–1467, 2002.

[2] Y. Eto and M. Suzuki, "Mathematical formula recognition using virtual link network," in *Proc. ICDAR*, 2001, pp. 762–767.

[3] Chen Y and Minoru O., "Structure analysis and semantic understanding for offline mathematical expressions.," in *International Journal of Pattern Recognition and Artifical Intelligence*, 2001, vol. 15, pp. 967–987.

[4] R. Anderson, *Syntax-Directed Recognition of Hand-Printed Two-Dimensional Mathematics*, Phd Thesis, Harvard University, Cambridge, MA, USA, 1968.

[5] Greg Mori, "Maching with shape context," in *http://www.eecs.berkeley.edu/Research/Projects/CS/vision/shape/sc-digits.html*. 2001, Computer Vision Group, Berkeley, University of California.

[6] C. Kam-Fai and Y. Dit-Yan, "Mathematical expression recognition: A survey," *International Journal on Document Analysis and Recognition*, vol. 3, pp. 3–15, 2000.

[7] M. Okamoto and B. Miao, “Recognition of mathematical expressions by using the layout structures of symbols,” in *Proc. ICDAR ’91*, 1991, pp. 242–250.

[8] F. Garvan, “ The MAPLE Book,” 2001.

[9] D. Blostein and A. Grbavec, “Recognition of Mathematical Notation,” *Handbook of Character Recognition and Document Image Analysis*, pp. 557–582, 2001.

[10] C. Faure and Z. Wang, “Automatic Perception of the Structure of Handwritten Mathematical Expressions,” *Computer Processing of Handwriting*, pp. 337–362, 1990.

[11] Xue-Dong Tian, Hai-Yan Li, Xin-Fu Li, and Li-Ping Zhang, “Research on symbol recognition for mathematical expressions,” *Innovative Computing ,Information and Control, International Conference on*, vol. 3, pp. 357–360, 2006.

[12] Richard G. Casey and Eric Lecolinet, “A survey of methods and strategies in character segmentation,” *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 18, no. 7, pp. 690–706, 1996.

[13] Nakano Y. Fujisawa, H. and K. Kurino, “Segmentation methods for character recognition:from segmentation to document structure analysis,” *Proceedings of the IEEE*, vol. 80, pp. 1079–1092, 1992.

[14] Y. Lu, “Machine printed character segmentation: An overview,” *Pattern Recognition*, vol. 28, no. 1, pp. 67–80, 1995.

[15] M. Koschinski, H.J. Winkler, and M. Lang, "Segmentation and recognition of symbols within handwritten mathematical expressions," in *Acoustics, Speech, and Signal Processing, ICASSP-95., 1995 International Conference on*, 1995, vol. 4.

[16] T. Due, A.K. Jain, and Taxt. T, "Feature extraction methods for character recognition-A survey," *Pattern Recognition*, vol. 29, pp. 641–662, 1996.

[17] HM Twaakyondo and M. Okamoto, "Structure analysis and recognition of mathematical expressions," in *Document Analysis and Recognition, Proceedings of the Third International Conference on*, 1995, vol. 1.

[18] M Okamoto and H.Msafiri, "Mathematical Expression Recognition by Projection Profile Characteristics," in *Trans. IEICE Japan*, 1995, pp. 366–370.

[19] S. Smithies, *Freehand Formula Entry System*, Master's Thesis, University of Otago, Dunedin, New Zealand, 1999.

[20] Serge Belongie, Jitendra Malik, and Jan Puzicha, "Shape context: A new descriptor for shape matching and object recognition," in *In NIPS*, 2000, pp. 831–837.

[21] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape context," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 24, pp. 509–522, 2002.

[22] F Yates, "Contingency table involving small numbers and the chi square test," vol. 1, pp. 217–235, 1934.

[23] I. Phillips, "Methodologies for Using UW Databases for OCR and Image Understanding Systems," *Document Recognition V*.

[24] Z.X.Wang and C.Faure, "Structural Analysis of Handwritten Mathematical Expressions ," in *In Proceeding of the nineth international Conference on Pattern Recognition*, 1988, pp. 32–34.

Vita

Ling Ouyang was born in Wuhan, China on November 28, 1984, the son of Xiao Ouyang and Lifang Yuan. He was raised in a big city, Wuhan, and attended high school in the Jianghan district. He enrolled in the BS Control Science and Technology program of Huazhong University of Science and Technology, China, in 2002 and the MS in Imaging Science program of Rochester Institute of Technology, Rochester, NY, USA in 2007. Ling interned with Xerox in Webster, NY in summer 2008 and currently is a full-time employee of Xerox as an image quality engineer.

Permanent address: 166 East Squire Dr, Apt 7,
Rochester, NY, USA.

This dissertation was typeset with LaTeX[†] by the author.

[†] LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.