CHINESE CALLIGRAPHIST: A SKETCH BASED LEARNING TOOL FOR LEARNING WRITTEN CHINESE

A Thesis

by

ZHENGLIANG YIN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Tracy Hammond
Committee Members,	Jim Xiuquan Ji
	Yoonsuck Choe
Head of Department,	Nancy M. Amato

December 2014

Major Subject: Computer Science

ABSTRACT

Learning Chinese as a foreign language is becoming more and more popular in western countries, however it is also very hard to be proficient, especially in writing. The involvement of the teachers in the process of learning Chinese writing is extremely necessary because they can give timely critiques and feedbacks as well as correct the students' bad writing habits. However, it is inadequate and inefficient of the large class capacity therefore it is urgent and necessary to design a computer-based system to help students in practice Chinese writing, correct their bad writing habits early, and give feedback personally.

The current written Chinese learning tools such as online tutorials emphasize writing rules including stroke order, but it could not provide practicing sessions and feedback. Hashigo, a novel CALL (Computer Assisted Language Learning) system, introduced the concept of sketch-based learning, but it's low level recognizer is not proper for Chinese character domain.

Therefore in order to help western students learn Chinese with better understanding, we adopted LADDER description language, machine learning techniques, and sketch recognition algorithms to improve handwritten Chinese stroke recognition rate.

With our multilayer perceptron recognizer, it improved Chinese stroke recognition accuracy by 15.7% than the average of the four basic recognizer. In feature selection study we found that the most important features were "the aspect of the bounding box", and the "density metrics", and "curviness". We chose 8 most important features after the recursive

selecting stabilized. We discovered that in most situations, feature recognition is more important than template recognition. Since the writing technique is emphasized while they are taught, only 2 templates is enough. It worked as well as 20 templates, which improved recognition speed dramatically.

In conclusion, in this thesis our contribution is that we (1) proposed a natural way to describe Chinese characters; (2) implemented a hierarchical Chinese character recognizer combining LADDER with the multilayer perceptron low level recognizer; (3) analyzed the performance of different recognition schemes; (4) designed a sketch-based Chinese writing learning tool, Chinese Calligraphist; and (5) find the best feature combination to recognize Chinese strokes while improving the recognition accuracy.

DEDICATION

I would like to dedicate this thesis to the member of my family: my mother Lijun Fan for her caring, my father Fuhua Yin for his constantly support and suggestion, my sister Fangming Yin who always shares news with me and keeps me in touch with my family, and my girlfriend Yunting Tan since she always cheers me up and gives me positive energy.

ACKNOWLEDGEMENTS

I would like to thank my supervisor Tracy Hammond, and my committee member, Yoonsuck Choe, and Jim Xiuquan Ji, for their guidance and support throughout the course of this research. Without their support and advices, it would not be completed.

Thanks also go to all the SRL lab members for making my time at Texas A&M University a great experience. My thanks go to Paul Taele for his help and direction guidance for the research topic. Thanks go to Stephanie Valentine, Hong-Hoe Ayden Kim, Manoj Prasad, Folami Alamundun, Jung In Koh, David Turner, Vijay Rajanna, Larry Powell, and Raniero Lara-Garduno for the lab services. Thanks to Andrew King III, Nahum Villanueva, Emily Berk, and Nick Henamann, which we had a lot of fun learning writing Chinese. Thank you all.

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iv

ACKNOWLEDGEMENTS v

LIST OF FIGURES viii

LIST OF TABLES xi

1. INTRODUCTION 1

2. RELATED WORKS 3

- 2.1 Computer Aided Language Learning in Writing Chinese Characters 3
- 2.2 Sketch Recognition 4
  - 2.2.1 Feature Based Recognizer 4
  - 2.2.2 Geometry Based Recognizer 8
  - 2.2.3 Template Based Recognizer 14
  - 2.2.4 Hierarchical Structured Recognizer 17
- 2.3 Machine Learning 22
  - 2.3.1 Neural Networks 22
  - 2.3.2 WEKA 23

3. METHODOLOGY 25

- 3.1 Chinese Character 25
  - 3.1.1 Chinese Character Structure 25
  - 3.1.2 Feature of Chinese Characters and Strokes 30
- 3.2 Chinese Stroke Recognition 34
  - 3.2.1 Choosing Recognizer: Gesture, Templates, or Geometry 34
  - 3.2.2 MultiLayer Perceptron Recognizer 36
  - 3.2.3 Preprocessing 39
- 3.3 Chinese Character Recognition 41
- 3.4 Error Detection and Feedbacks 43

4. IMPLEMENTATION 45

- 4.1 System Architecture 45
- 4.2 Stroke and Shape Data Structures 46

4.3 Feedback Contents 48
4.4 Interface Design 51

5. EXPERIMENTS AND EVALUATION 56

5.1 Data Collection 56
5.1.1 Sousa 56
5.1.2 Choosing Input Method 57
5.2 Evaluation Design 57
5.3 User Study Design 59
5.4 Results 60
5.4.1 Stroke Recognition Accuracy 60
5.4.2 Character Recognition Result Analysis 67
5.4.4 Feedback Result Analysis 69
5.5 Feature Selection 70

6. FUTURE WORKS 74

7. CONCLUSIONS AND CONTRIBUTIONS 75

REFERENCES 77

APPENDIX A 80

APPENDIX B 84

LIST OF FIGURES

Page

Figure 1 Previous sketch based interfaces for language learning 3

Figure 2 Rubine feature examples...5

Figure 3 Sezgin corner finding example 9

Figure 4 Direction, curvature, and speed graph of a stroke 9

Figure 5 Threshold setting in sezgin vertex finding..10

Figure 6 Line test, arc test, and direction graph in primitive recognition 11

Figure 7 Shape approximation and beautification...12

Figure 8 Examples of corners and straws in shortstraw...................................14

Figure 9 Resampling and rotating examples used by One Dollar recognizer 15

Figure 10 Definition of character “Write” by LADDER 18

Figure 11 Definition of radical “Cover” by LADDER with Chinese stroke primitive....19

Figure 12 Definition of radical “Cover” by LADDER with geometric primitive 20

Figure 13 Example of the compound stroke “ShuZheWanGou” 21

Figure 14 The evolution of the shape of Chinese character “Water”...............25

Figure 15 Chinese character, radical, stroke hierarchy 27

Figure 16 Example of character “Good” in hierarchical view 27

Figure 17 List of 8 basic chinese strokes 27

Figure 18 List of 29 compound chinese strokes...28

Figure 19 Chinese character raw frequency in 200 million character corpus 31

Figure 20 Related cumulative frequency in percentile of Chinese character....................31
Figure 21 Strokes used in Chinese Calligraphist 32
Figure 22 Features and neural networks structure of designed MLP recognizer 36
Figure 23 Reampling algorithm adopted...40
Figure 24 Comparison of stroke shapes between before and after preprocessing 41
Figure 25 LADDER definition of radical “Water” in XML view 42
Figure 26 LADDER definition of character “Again” in XML view................................42
Figure 27 LADDER definition of character “Chinese” in XML view 43
Figure 28 LADDER feedback design of radical “Water” in XML view 44
Figure 29 Interface data flow and control flow 45
Figure 30 The UML of class stroke 47
Figure 31 The UML of class DomainShape...49
Figure 32 Overview of the Chinese Calligraphist GUI..51
Figure 33 Problem information panel 52
Figure 34 Character stroke guiding and feedback panel 52
Figure 35 Input and sketch panel 52
Figure 36 Interactive control buttons 52
Figure 37 Text feedback panel 53
Figure 38 Confirming correct writing feedback...54
Figure 39 Warning incorrect writing feedback 54
Figure 40 Dispalying complete and correct writing feedback 55
Figure 41 Text feedbacks about the recognized strokes, radicals, and characters 55

Figure 42 Wacom bamboo device used in collecting training and testing data 56

Figure 43 Comparison of f-measures among different recognition schemes...................67

Figure 44 Example of unrecognized characters in user study..68

Figure 45 Compaison of f-measure for various template numbers and features 71

LIST OF TABLES

Page

Table 1 Explanation and formula of Rubine features..6

Table 2 Explanation of Long features 7

Table 3 Correlated Long feature dimensions 7

Table 4 Hierarchy comparison between Chinese and English 29

Table 5 Hierarchy example of "wisedom" between Chinese and English.......................29

Table 6 Chinese stroke similar groups 33

Table 7 Comparison among different types of recognizers for Chinese domain 35

Table 8 Contents to check in different types of feedbacks 50

Table 9 Survey questions for feedbacks of the interface after user study 60

Table 10 Confusion matrix for Rubine recognition result 61

Table 11 Confusion matrix for Long recognition result 62

Table 12 Confusion matrix for Hausdorff recognition result...63

Table 13 Confusion matrix for OneDollar recognition result 64

Table 14 Confusion matrix for MultiLayer perceptron recognition result.......................66

Table 15 Feature selection results 73

1. INTRODUCTION

Learning a foreign language involves speaking, listening, reading, and writing. So does Chinese. As Chinese becomes more and more popular among western students, the challenges arise to them because of the different language systems. Currently lots of techniques are developed to help western students with speaking and reading Chinese, with the plenty of online resources generated due to the internet booming, such as online courses, YouTube videos and shows. Ironically, when it comes to writing, it is a different story even though the writing is much harder than reading due to the difference between morphemic and phonemic system. There are very few tools to help the students improve writing skills.

Traditional classes learning Chinese writing for native students involves lively story and history about the how character changes to the shape nowadays. Students remember the meaning of the shape with fun. Besides, they have lots of practice, and feedbacks either from the teachers or their family. Writing techniques are emphasized as well since it is important to build their understanding of how the characters are organized, meanwhile it builds the idea of indexing in dictionary.

Different from native Chinese students learning the writing as their first language, the western students spend much less time. Moreover, there are less chance for them to get feedbacks of their writing due to the limits from teachers, TAs, and tutors, as well as the complete written characters rather than online writing process. Even worse, teachers don't tell the fascinating story behind each character of how it becomes the shape. Instead, it has been reduced to a solely sequence of lines to memorize and write over and over again, which takes away all the fun, mystery, and all thousands of years of history. Less practice, less feedbacks, and less fun, makes it is very necessary to develop a Computer Aided Language Learning (CALL) in order to compensate the insufficiency of the class. However current online learning tools are limited to basic canvas functionality.

Hashigo is an available CALL with feedback functionality. It checks the visual correctness and writing technique correctness of the sketches. But the problem is that it takes lines as recognition primitives. However, curving strokes are also important concept in Chinese language as well. Therefore, several improvements should be made upon the concept of Hashigo. Firstly, we should not using lines as primitives, even with curves. That's because the curving or polyline recognition is not convincingly stable by using the low level recognizer used in Hashigo. Secondly, it might break a single Chinese concept stroke into several parts, which breaks the concept.

This paper describes a new written Chinese learning tool based on sketch recognition, called Chinese Calligraphist, helping students learning Chinese writing, with providing practicing session, recognizing, and feedbacks. In this paper, we researched on Chinese character structure, designed a specifically Chinese stroke recognizer, and implemented LADDER to recognizer characters with the Chinese stroke primitive shapes. And we analyzed the performance of recognition as well as collected feedbacks about the interface and user experience after user study.

2. RELATED WORKS

2.1 Computer Aided Language Learning in Writing Chinese Characters

Kanji Storyteller [17] is a sketch-based interface for learning Kanji by Ross Peterson that tries to help students understand characters by combining the shape, image, and story of the character, shown as Figure 1(left). It is a nice supplement of a CALL, but, not offering practices and feedbacks to the students, which are the main parts of learning writing.

Hashigo is also a sketch based interactive system for Kanji, it assesses both the visual structure and written techniques of students' writing [22], shown as Figure 1(right). It adapts free-sketch recognition techniques and offers both learn mode and review mode. Hashigo adopts geometric based primitive shape recognizer [15] as low level to recognize primitive sketches such as lines, curves, and ellipses. And then combining them with some constraints under the language domain definition of high level character shapes.

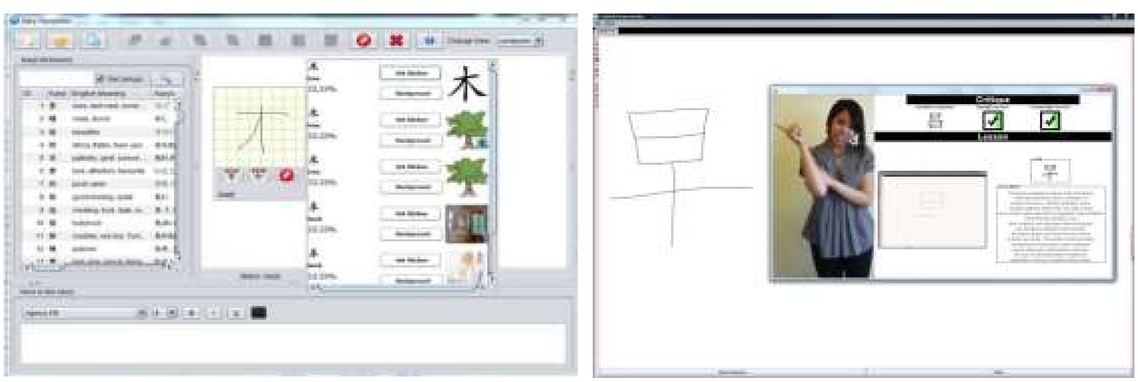


Figure 1 Previous sketch based interfaces for language learning

However, since the curve and polyline recognition is not absolutely stable in low level recognition, and for curve the low level recognize would not return the differences among the curve variations, plus the curves are not as informative as lines, when dealing with characters writing, the assumption made by Paul prefers the lines rather than curves. However, this is not proper and intuitive, because the slightly curving strokes has their own meaning. Especially for the students are learning the concept, it would be misleading.

2.2 Sketch Recognition

Sketch Recognition is the automated recognition of hand-drawn diagrams by a computer. In other words, that is to make computer understand what human's intention of drawing. In this section, we will briefly overview some handwritten recognition algorithms and techniques developed in the past years. We focus on online recognition algorithms, which implies a natural way of interaction between human and computer. From recognizing gestures, to domain independent primitive shapes, and to matching templates, the characteristics and applications of the algorithms are both varying a lot.

2.2.1 Feature Based Recognizer

2.2.1.1 Rubine

Rubine recognizer is a single stroke gesture recognizer proposed by Dean Rubine [18]. The single stroke simplifies the system since it avoids segmentation problem. It is rapidly trained from a small number of examples. Rapid training time is significant for a system which is used for prototyping gesture-based systems, because it makes the designer of the system to easily experiment with different sets of gestures for a given application. [19]. D. Rubine presents his feature sets, showing in Table 1, as Figure 2. Under his logic, the criteria for selecting features are incrementally computability, meaningfulness, and enough but not excessive amount.

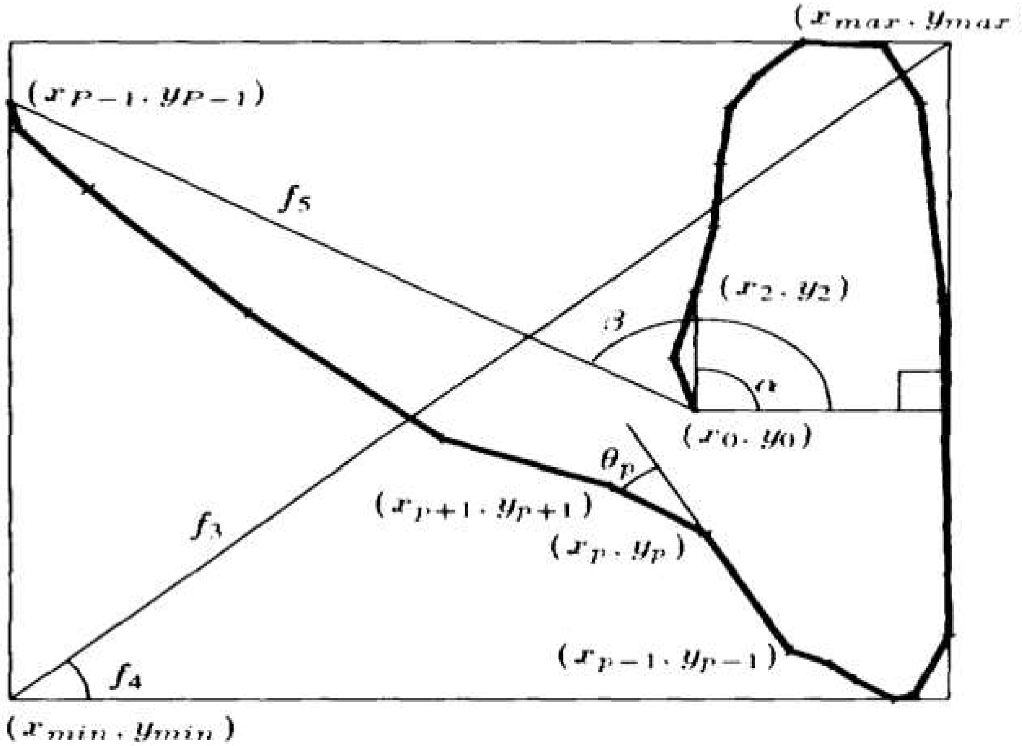


Figure 2 Rubine feature examples

With these features, train the recognizer with labeled sample data, and apply it with a linear classifier, and rejection control, it becomes the basis of gesture recognition system.

2.2.1.2 Long

In 2000, Long described an improved model to distinguish the similarities among gestures for the pen based UI systems. In the studies [13], he did two separate gesture similarity experiments. The algorithm is derived from Rubine, which he selected the first 11 features, and added 11 new more, as shown in Table 2.

Table 1 Explanation and formula of Rubine features

Feature #	Comment	Formula
1	Cosine of the initial angle with respect to X axis	$f_1 = \cos\alpha = (x_2 - x_0)/d$ $d = \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}$
2	Sine of the initial angle with respect to X axis	$f_2 = \sin\alpha = (y_2 - y_0)/d$ $d = \sqrt{(x_2 - x_0)^2 + (y_2 - y_0)^2}$
3	Length of the bounding box diagonal	$f_3 = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}$
4	Angle of the bounding box	$f_4 = \tan^{-1}\left(\frac{y_{max} - y_{min}}{x_{max} - x_{min}}\right)$
5	Distance between first and last point	$f_5 = \sqrt{\left(x_{p-1} - x_0\right)^2 + (y_{p-1} - y_0)^2}$
6	Cosine of angle between first and last point	$f_6 = \cos\beta = (x_{p-1} - x_0)/f_5$
7	Sine of angle between first and last point	$f_7 = \sin\beta = (y_{p-1} - y_0)/f_5$
8	Total gesture length	$\Delta x_p = x_{p+1} - x_p \quad \Delta y_p = y_{p+1} - y_p$ $f_8 = \sum_{p=0}^{P-2} \sqrt{\Delta x_p^2 + \Delta y_p^2}$
9	Total angle traversed	$\theta_p = \tan^{-1}\left(\frac{\Delta x_p \Delta y_{p-1} - \Delta x_{p-1} \Delta y_p}{\Delta x_p \Delta x_{p-1} + \Delta y_p \Delta y_{p-1}}\right)$ $f_9 = \sum_{p=1}^{P-2} \theta_p$
10	Total absolute angle traversed	$f_{10} = \sum_{p=1}^{P-2} \lvert\theta_p\rvert$
11	Total squared angle traversed, Sharpness	$f_{11} = \sum_{p=1}^{P-2} \theta_p^2$
12	Maximum speed (squared)	$\Delta t_p = t_{p+1} - t_p$ $f_{12} = \max_{0 \le p \le P-2} \left(\left(\Delta x_p^2 + \Delta y_p^2\right)/\Delta t_p^2\right)$
13	Path duration	$f_{13} = t_{p-1} - t_0$

Table 2 Explanation of Long features

1	**Cosine of initial angle**	**12**	**Aspect [abs(45-#4)]**
2	**Sine of initial angle**	**13**	**Curviness**
3	Size of bounding box diagonal	**14**	Total angle traversed / total length
4	**Angle of bounding box**	**15**	**Density metric 1 [#8/#5]**
5	**Distance between first and last points**	**16**	**Density metric 2 [#8/#3]**
6	**Cosine of angle between first and last points**	**17**	**Non-subjective "openness" [#5/#3]**
7	**Sine of angle between first and last points**	**18**	Area of bounding box
8	Total length	**19**	Log(Area)
9	**Total angle**	**20**	Total angle / total absolute angle
10	**Total absolute angle**	**21**	**Log(total length)**
11	**Sharpness**	**22**	**Log(aspect)**

After experiments, Long analyzed how people decided similarities among features in multiple dimensions. The result is that the optimal dimension is five, shown as Table 3. Most significantly, he found that a small number of features explain the most salient dimensions: neither length nor area were significant, and the log of aspect had more influence on similarity than aspect itself.

Table 3 Correlated Long feature dimensions

Dimension	**Correlated features (In order of descending importance)**
1	Curviness, Angle / distance
2	Total absolute angle, Log(aspect)
3	Density 1, Cosine of initial angle
4	Cosine of angle between first and last points, Cosine of initial angle, Sine of initial angle, Distance between first and last points, Angle of bounding box
5	Aspect, Sharpness, Cosine of initial angle, Total angle

2.2.2 Geometry Based Recognizer

Specifying gestures is a very efficient way to recognize hand drawn sketches so that the system could react seamlessly. However, it have many constraints. One is that it performs poorly if there are many gestures or gestures are similar. Another limitation is that gestures differ person to person, therefore for the same purpose, it could be thousands of gestures just for the only one meaning, and even the same gesture could have different semantic meanings, which requires too many redundant, tedious, and inefficient training. On the other hand, geometry based recognizer are more focused on recognizing the geometrical features and extract some regulations so that it could apply as subshapes to any different hyper complex shapes.

2.2.2.1 Sezgin

Different from previous systems specifying gestures, the system presented by Sezgin is dealing with vertices and geometric knowledge to build graphics [20]. It tries to recognize the gestures or shapes more like what a human do. The task is to understand what people are drawing by accurate early processing the basic geometry-finding corners, fitting both lines and curves. For example, in Figure 3, human can detect four corners of one square polyline stroke.

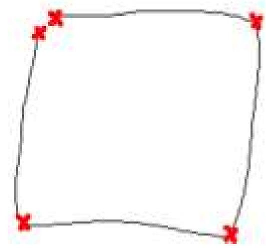


Figure 3 Sezgin corner finding example

Sezgin system has three phases in processing: approximation, beautification, and basic recognition. Approximation fits the most basic geometric primitives to a set of pixels, which is the most important; beautification modifies the output of the approximation layer, primarily to make it visually more appealing without changing its meaning; basic recognition produces interpretations of the strokes, like interpreting a sequence of four lines as a rectangle or square.

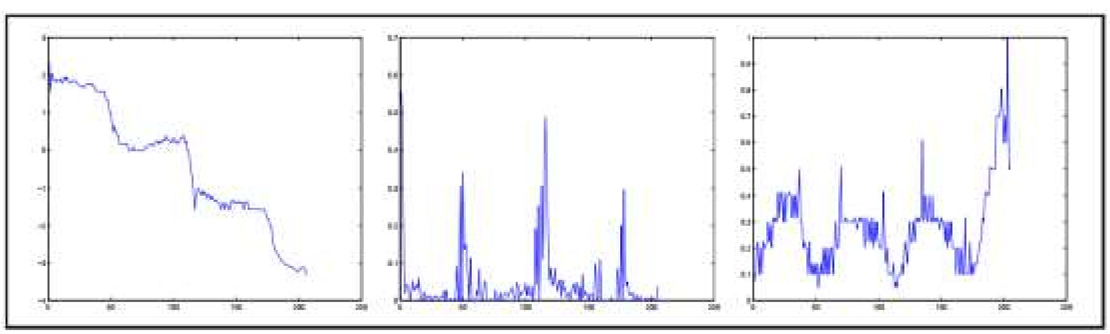


Figure 4 Direction, curvature, and speed graph of a stroke

In the process of stroke approximation, the first step is to detect vertex or corner by looking for points along the stroke, that are minima of speed and maxima of the absolute value of curvature, shown in Figure 4. To find this extremes, average based filtering is used, and to avoid the problems posed by choosing a fixed threshold, the threshold are set as the mean of each data set, shown as Figure 5. Therefore it is dynamic. Specially, for curvature data, the threshold is the mean, while for the speed the threshold is ninety percent of the mean. For curvature and speed data alone, they may have some problems on each dataset. Only for combining them together, it can reduce as many as possible the false positives. The hybrid fit generation occurs in three stages: computing vertex certainties; generating a set of hybrid fits; and selecting the best fit. The initial hybrid fit H_0 is the intersection of speed minima corner fit and curvature maxima corner fit. Then on each cycle augment it with best speed or curvature candidates.

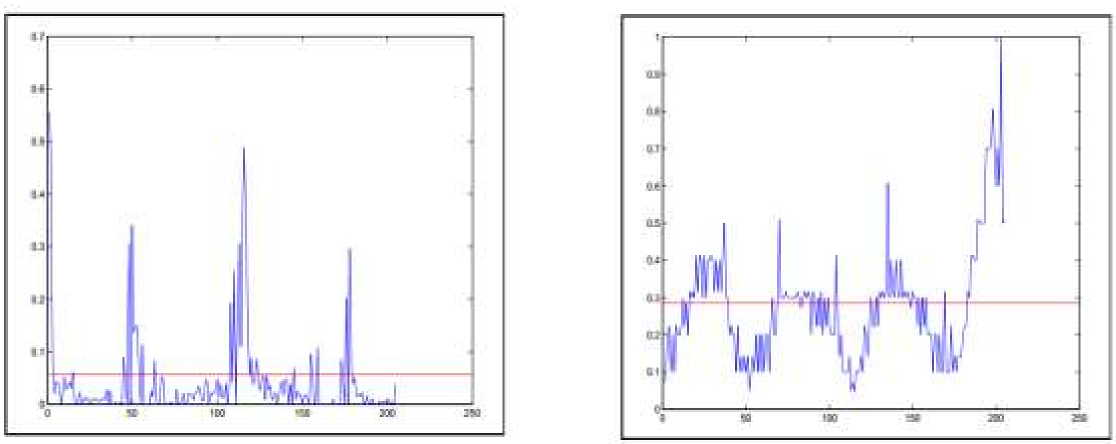


Figure 5 Threshold setting in sezgin vertex finding

For basic recognition, the first step is to determine whether they are curves or lines. If the Euclidean distance is significantly smaller the stroke length, it shall be recognized as curves. And then compute the control points of the curves as Bezier curve. If the error is large enough, then recursively divide it into two halves and repeat the process until it is proper processed. Finally beautify the shape with the best fit lines or Bezier curves.

Sezgin corner finding works with more complicate domain or systems since it is not a template based or Rubine feature based recognizer. It is able to use multiple sources of information to produce good approximations of freehand sketches.

2.2.2.2 Primitive Recognizer

For domain independent sketch recognition, the low level primitives are essential parts, which include line, arc, circle, ellipse, spiral, etc. This shapes can be recognized with no knowledge of high level domain information.

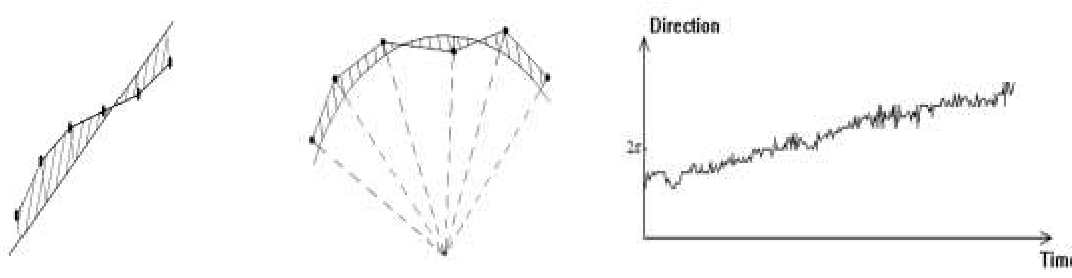


Figure 6 Line test, arc test, and direction graph in primitive recognition

In 2003, Yu and Cai proposed their interface allowing users to draw their sketches as natural as in paper so that it would not harm the user experience of human computer interaction. For the stroke approximation, the most significant stats are the direction and curvature [25]. Plus feature area of different shapes, examples as Figure 6. If they meet the requirements and the constraints, then the system would translate the original stroke to the approximated stroke. Figure 7 shows the circle and star polyline approximation respectively.

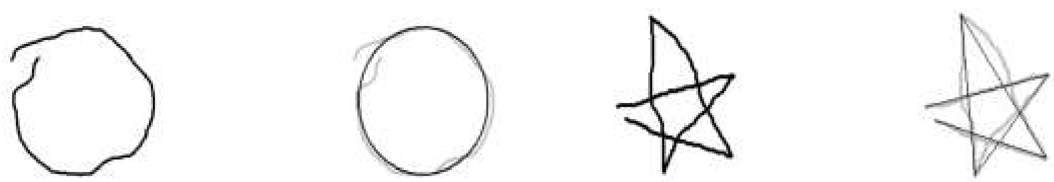


Figure 7 Shape approximation and beautification

2.2.2.3 PaleoSketch Recognition

Based on the Sezgin corner finding techniques, as well as Yu and Cai primitive shapre recognition system, Paulson and Dr. Hammond improved the low level primitive recognizer with novel preprocessing steps, which is called PaleoSketch recognizer [15]. The new recognizer compute two new features called NDDE and DCR during the pre-recognition process in order to aid distinguish curve from polyline. NDDE stands for normalized distance between direction extremes, which is computed as the distance between highest and lowest direction divided by distance of the whole stroke. It is useful to determine how shape of angle or stroke is. If it is a polyline or angle, it shall be very small, close to 0, because of the direction extremes transmission complete in a short length. On the other hand, if it is a curve or circle, it shall be very high to 1 since the total rotation is finished gradually with the whole stroke. DCR, direction change ratio, is calculated as max direction change divided by average direction change. This indicates the change speed of the direction. Polylines has this value very high, and ellipse, arcs, circles have this value close to 1. Another novel features of the recognizer is that it gives out the confidence of different possible shapes, including hierarchy shapes such as complex. For different interpretations, the confidence or scores are calculated by the combination of the scores of the basic shapes. For line is 1, arc is 3, and curve, circle, ellipse, helix, and spiral are all 5. Although the confidence is not that accurate, however, it offers the possibility to maintain and rank the list of interpretations. For a more accurate recognition decision, the neural network version of the PaleoSketch was brought up later [16].

2.2.2.4 Other Corner Finding Algorithms

Corner Finding is a key step and part in geometry based sketch recognition. Both in Sezgin, Yu, and Paleo, it first detects the corners and breaks the stroke to substrokes, and fit the substrokes into line, arc, or curve and other shapes. So that complex shape could be build up with the precisely recognized parts.

In 2008, Wolin also presented a novel and easy-to-implement corner finder algorithm called ShortStraw [24]. A straw for point pi is $straw_i = |p_i\text{-}w,\ p_i\text{+}w|$, euclidean distance, where the w is the window, shown in Figure 8. The steps of the shortstraw corner finding is resampling, calculating the "straw" around each resampled point, and taking the points with the minimum straw distance to be corners. Afterwards, the top-down stage finds missing corners and removes points by checking consecutive corners.

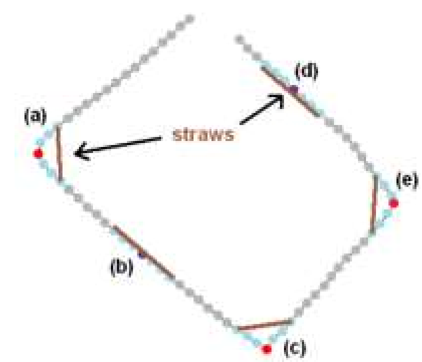


Figure 8 Examples of corners and straws in shortstraw

2.2.3 Template Based Recognizer

Besides the recognizers with calculating features or geometric constraints, another intuitive way to recognize sketches is template matching. It has both advantages and disadvantages. The benefits from template based recognition is apparent. It needs no learning and training. The whole recognition could be template-and-go. Once there are the templates of gestures to be recognized, and match the points with several preprocessing steps, and the recognition results will be returned. On the other hand the disadvantage is also obvious, that is it needs to compare an unknown gesture with all of stored templates to make a prediction so that it may be both time and space consuming.

2.2.3.1 One Dollar Recognizer

In 2007, Jacob presented an interface to recognize and to create gestures based on the template matching. It is not complicated and called one dollar recognizer [23]. The steps are natural: resample the points into N evenly spaced; rotate points so that their indicative angle is at 0°; scale points to the fixed size and translate the points to the origin; and finally match points against a set of templates. Figure 9 shows the example of the first two steps, resample and rotation.

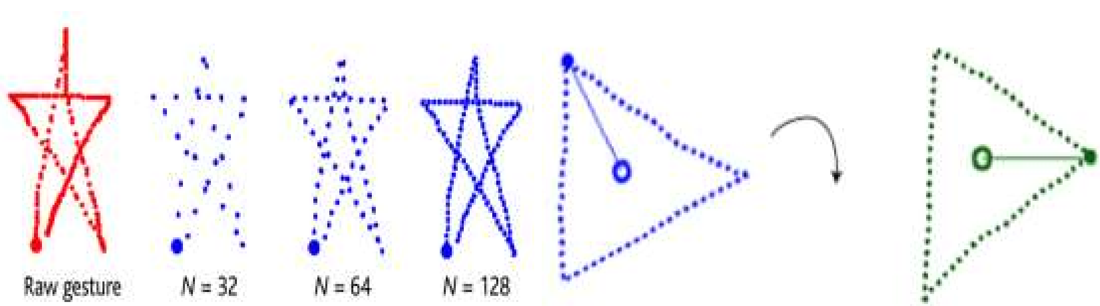


Figure 9 Resampling and rotating examples used by One Dollar recognizer

To be noted, the resample step is not based on the points' density, but the evenly distributed distance. Also, for general gestures, different number of resampling points is also important. It should not be too wide, since we will lost the shape information. Neither it should be too dense, otherwise the complexity is increased sharply. To be fairly, 64 is enough and yields good performance in practice.

To improve the performance and solve the problem of disadvantage: comparing too many templates, Protractor [10] was proposed by Yang Li in 2010. It uses the method of nearest neighbor. For each gesture, either unknown or training sample, Protractor preprocesses it into an equal length vector. It also uses a novel closed-form solution to calculate an optimal angular distance in order to improve the accuracy and speed. The most aspect it improves from one dollar is in handling the orientation sensitivity and scaling.

2.2.3.2 Hausdorff Recognizer

Hausdorff is presented by Kara [9] and it has been employed in many different places including an engineer course oriented recognition system [3] [4], which is originally used to recognize “truss”. Before calculate the hausdorff distance, it computes two distance vectors for two shapes $D_A = \left\{ \min_{b \in P_B} (|a - b|), a \in P_A \right\}$. The distance vector is that for each point in one shape, find the nearest point in the other, and the distance between them represents the distance of the former one from the first shape to the second. And the hausdorff distance between two shapes is defined as the maximum of the distance in the distance vectors of the two shapes $H_d(A, B) = \max\left(\max(D_A), \max(D_B)\right)$.

The modified hausdorff distance is the average of the minimum distances for all points from its shape to another. $H_{mod}(A, B) = \frac{\overline{D_A} + \overline{D_B}}{2}$. We need to convert from the distance to the confidence or probability of the shape. $P(match) = 1 - H/20$, where 20 is a threshold representing half of the bounding box size.

2.2.4 Hierarchical Structured Recognizer

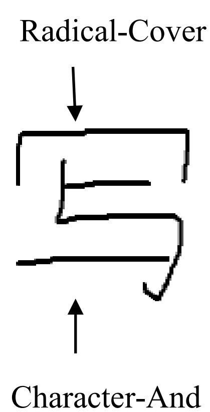
Hierarchical sketch recognition architecture is consist of low level primitive recognition and high level complex recognition. The low level primitives can not only be domain independent such as line, arc, curve, circle, etc., but also be some domain dependent primitive elements such as Chinese radicals or basic strokes. But the high level complex recognition introduces the necessity of methods to recognize the combination of low level primitive shapes in some constraint way. LADDER is the tool we use which provides the possibility to describe and recognize constraints. This will be discussed in the following section.

2.2.4.1 LADDER

LADDER stands for Language to Describe, Display, and Editing in Sketch Recognition (LADDER) proposed by Hammond [6] [7]. The use of LADDER is to group the primitives using pattern recognition techniques. There are several applications of using LADDER. Paul in his thesis employed LADDER to recognize symbols from various EA writing scripts [21] [22]. Alvarado designed a multi-domain sketch recognition engine by adopting LADDER for domain knowledge constraints description tool as well [2]. The domain description in LADDER describes the shapes with three aspects: components, constraints, and alias. For components, it describes the sub shapes consisting the shape. It could be either compound shapes, or primitive shapes. Such as in Figure 10, it shows the Chinese character "write", in the right part, the components contains a radical and a character, which are both compound shapes. However, the components of shape radical "cover" could be described as Lines, which are the primitive shape recognized by lower level recognizer.

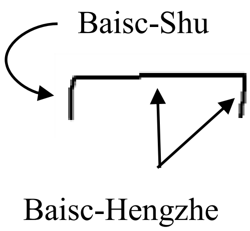
Figure 10 represents the Chinese character ten. The components field shows that the character ten should have two lines: a horizontal line and a vertical line. The constraints and aliases fields represent the geometric rules for the two lines.

Figure 11 and 12 shows the ladder definition of Chinese Radical "cover" in two different ways. In Figure 11, it is represented as two basic strokes in Chinese basis stroke set, this is intuitive and straightforward as students learning Chinese characters, and aiding students to understand the structure of Chinese characters better.



Name:		
Character-Write		
Components:		
Radical-Cover		cover
Character-And		and
Constraints:		
Above	cover-Bottom	and-Bottom
Above	cover-Top	and-Top
Above	and-Top	cover-Bottom
SameX	cover-Left	and-Left
SameX	cover-Right	and-Right

Figure 10 Definition of character "Write" by LADDER

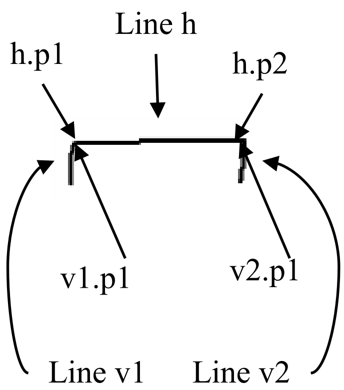


Name:		
	Radical-Cover	
Components:		
	BasicStroke-Shu	shu
	BasicStroke-HenZhe	hengzhe
Constraints:		
Conincide	shu.p1	hengzhe.p1
Alias:		
	shu.p1	shu.topmostPoint
	hengzhe.p1	hengzhe.leftmostPoint

Figure 11 Definition of radical "Cover" by LADDER with Chinese stroke primitive

In Figure 12, the radical "cover" is represented as three lines, and way much more constraints than the first definition. This could be more efficient in some scenarios, especially considering only recognize the looking correctness. This way it would be more likely to return positive feedbacks to the user since the looking of their writing is correct. However, when the canvas becomes messy and too many strokes to consider and segmentation, the test complexity grows exponentially.

Since the basic strokes are not too complex, if the basic strokes could not be recognized, it is also more likely to write the characters wrongly. Therefore to obey the rigid hierarchical structure does not reduce the feedback quality if the looking is correct.



Name:		
	Radical-Cover	
Components:		
	Line	vline1
	Line	vline2
	Line	hline3
Constraints:		
Conincide	vline1.p1	hline.p1
Conincide	vline2.p1	hline.p2
SameLength	vline1	vline2
Horizontal	hline	
Vertical	vline1, vline2	
LeftOf	vline1	vline2
Alias:		
	vline1.p1	vline1.topmostPoint
	vline2.p1	vline2.topmostPoint
	hline1.p1	vline1.leftmostPoint
	hline1.p2	vline1.rightmostPoint

Figure 12 Definition of radical “Cover” by LADDER with geometric primitive

Should we consider the basic strokes as primitives or high level shapes? There are two ways to interpret the basic strokes. One way is to consider them as high level shapes, and they should be recognized from low level primitives and high level constraints. This way, it is consistent with the domain hierarchy structure. One the other hand, the basic stroke could be regarded as the primitive shapes themselves too. It needs the low level recognizer to recognize it. This could be only applied to template based or feature based classification. The logic behind it is that there are not too many basic strokes in the Chinese basic stroke set therefore the classification is proper since we are not try to classify them as characters which has thousands of classes and a very many similarities. Another benefit is that it could avoids some misinterpretations of the low level primitive strokes. For example, the “hengzhe” could be recognized as three polyline instead of two, which brings noise. However, it makes the low level recognition domain independent on the other hand.

Figure 13 Example of the compound stroke “ShuZheWanGou”

Let’s look at another example. One stroke in the Character “And” is shown in Figure 13 on the right side. It is called “ShuZheZheGou”, simplified as “SZZG”. If it is considered as high level strokes using basic geometric strokes, it should contain a polyline with four lines. One vertical line at top, and at the bottom of the vertical line, a horizontal line going towards right, then a nearly vertical line slightly goes down left, with finally a short hook line going upleft. When describing like this, it becomes tangent and no longer a correct way of learning writing, not to mention the noise and recognition error from low level recognizer. This will be discussed more in methodology chapter.

2.2.4.2 SketchRead

SketchRead [2] is an implementation of hierarchical recognition system. It adopts sezgin as for its low level recognition, and LADDER for high level domain description. It is focusing on circuit design graphics, and family tree graphs. The most significant and novel feature of the system is that it employs the Bayesian belief network, so it adjusts the confidence of interpretations according to the low level shapes and the prior probability of the high level shapes. And once a higher level shape is of high probability, it will consider other strokes as the rest parts of the shape with bias, which means searching for the missing parts actively. The integration of dynamically constructed Bayes nets [1] will be discussed in implementation chapter.

2.3 Machine Learning

Machine learning is widely applied in handwriting recognition. From Rubine classification to SketchRead presented above, almost each system applied more or less the concepts and techniques of machine learning. Classification is used to decide what class (gesture) classified according to the feature vectors extracted the sketch and the knowledge gained from training [18]; neural networks classification provides more self-adjusted way and precise confidence [8] [26]; hmm is also used in recognizing Chinese handwriting [11] [12] [14]; pattern recognition are introduced in LADDER, and baysian networks are employed in SketchREAD. In a nutshell sketch recognition is an important development and application of machine learning.

2.3.1 Neural Networks

Neural networks is supervised learning algorithm. It imitates the human brain function so that from input to output, nodes are connected with weights, which can be learned from training. Multi-layer neural networks even overcomes the limitation of single layer perceptron. The perceptron is still linear classifier, however, by introducing hidden layer units, the neural networks could learn the feature and adjusting the relationship between inputs, and it could classify with much higher accuracy and returns the confidence of the class.

In Taele's thesis, he argues that NN have strength in recognizing based on their visual structure however it brings weakness in assessing the writing techniques of the students since the timing and ordering are disregarded. It is true if it simply employs NN for recognizing the complete writing. However, with the hierarchy recognizer. If it just introduce NN to low level recognition, it would boost the accuracy of the primitive stroke or shape recognition vastly.

Therefore, when testing the performance of low level recognition for Chinese characters, instead of just using the original Rubine and Long, we introduced NN to them, more specifically, multi-layer perceptron back propagation. This discussion will be found in methodology chapter.

2.3.2 WEKA

WEKA is a tool for and collection of machine learning algorithms. WEKA, literally stands for Waikato Environment for Knowledge Analysis which provides both experimental UI and java API for systems integrating with machine learning such as classification, clustering, neural networks etc. It has been used in CALVIN [5] before, which is a sketch recognition system specific on COA (course of action) domain. As far as current knowledge, there are not too many CALL system employing WEKA. However, with WEKA, it would equip the system with much more ability of machine learning and would simplify the implementation.

3. METHODOLOGY

3.1 Chinese Character

3.1.1 Chinese Character Structure

As learning any other languages, learning Chinese for a non-native students is very difficult. The seemly only way to master it is to practice and to memorize all the shapes. Fortunately, Chinese characters are not random collection and combination of strokes and lines. On the other hand, behind the shapes, there are some patterns in constructing the characters.

Figure 14 The evolution of the shape of Chinese character "Water"

Chinese character is logogram, which is also hieroglyphs informally. That is to say, the current Chinese characters, both traditional, and simplified, especially their ancient symbol origin, is representing the shape, and sound describing the meaning of the character. For example, character "水", which means water, is derived and changed as Figure 14. At the beginning, the symbol is just representing the shape of creek and river, later on, with the unifying and merging different writing style, the character developed more rigidly and neatly. At the end, a unified and systematic writing standard was developed.

Chinese character is a "Stroke – Radical – Character" hierarchical structure, shown as Figure 15. Example of Figure 16 is how strokes become radicals, and radicals become characters. The character "好" means good, while "女" means women, and "子" means children. To combine together, a family with kids and women, indicating wife, is complete and happy, which is good.

Stroke is the most basic concept. In writing, it is an undividable unit. To classify different kinds of strokes, there are basic strokes, and compound strokes. For basic strokes, such as Heng , Shu , Pie , Na , there are 8 different kinds in total. They all can be written in one hand/pen movement naturally, except for Gou , which has a turn and hook at the end. However, they are all the simplest strokes in Chinese writing. For compound, or complex strokes, there are 29 various kinds all together. Most of them are in terms of combining some basic strokes. However, these strokes are also meant to finish in one writing stroke with some direction turns. Not all these 29 strokes are commonly used. For many starter learners, lots of daily and most used characters are consists of a subset of these 29 strokes. Note that also most of strokes have similarities with each other, shown as Figure 17 and Figure 18. These features will be discussed later in this section.

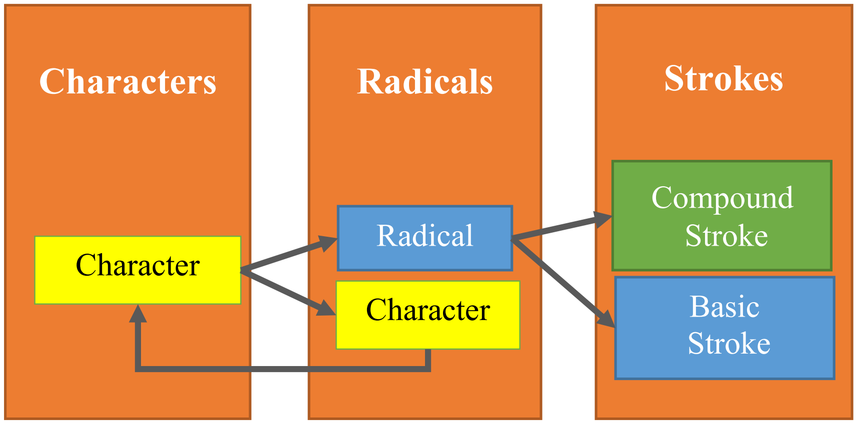


Figure 15 Chinese character, radical, stroke hierarchy

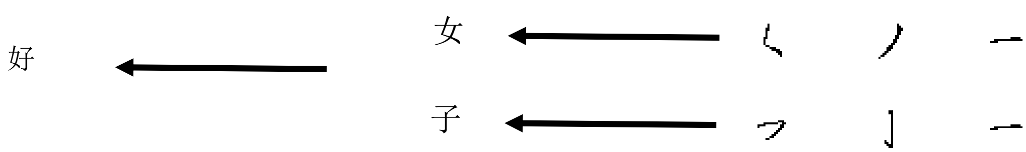


Figure 16 Example of character "Good" in hierarchical view

Figure 17 List of 8 basic chinese strokes

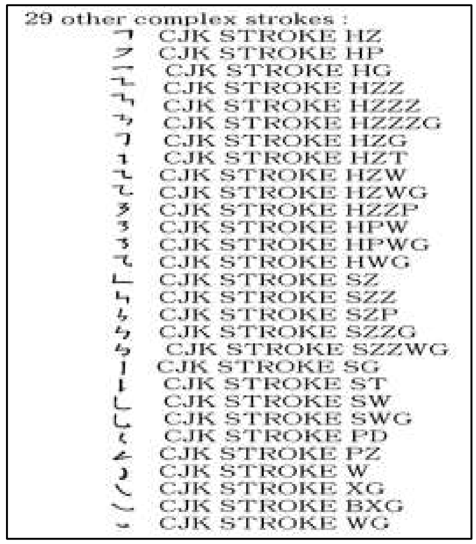


Figure 18 List of 29 compound chinese strokes

Radical is then an advanced concept. It combines several strokes, or rather sometimes can be a single stroke, to represent some meaning. It could either be a complete character, or a part of it. The radical in Chinese is a similar concept to the prefix, suffix, and root in English, if we compare Chinese strokes to English Latin letters. For example, the radical is the combination of strokes, and prefix/suffix is the combination of letters; both combinations are not random, but following the lexical rules: in English, the ordering of letters matters, do does the ordering and spacing of strokes in Chinese.

Character is the final concept as a word representing some meaning and a basic unit to form a sentence and paragraphs. Table 4 describes an analogue between English and Chinese, meanwhile Table 5 shows an example of the word meaning “wisdom” in both language. In English, the root of wisdom is wise, and –dom is to make it as a noun.

On the other hand, in Chinese, radical “知” means knowledge, to understand, and “日” means day, daily. Also to combine with, that daily accumulated knowledge is wisdom.

Table 4 Hierarchy comparison between Chinese and English

Chinese	
stroke	Finite, simple, similarity
radical	Combination of strokes, ordering, spacing
character	Combination of radicals/words, ordering, spacing
English	
letter	Finite, simple, similarity
root/radical/prefix/suffix	Combination of letters, ordering
word	Combination of root/radical/pre/suffix

Table 5 Hierarchy example of “wisedom” between Chinese and English

Chinese: 智	
stroke	㇒ ㇐ ㇔ ㇑ ㇕
radical	知 (knowledge)，日 (day)
character	智
English: Wisdom	
letter	W, I, S, D, O, M
root/radical/prefix/suffix	WIS (WISE), DOM (-DOM)
word	WISDOM

3.1.2 Feature of Chinese Characters and Strokes

Same as any language, Chinese characters express the Zipf's law in daily usage statistics as well. Zipf's law states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. For example, the most frequent word might occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, and so on. In our study, we checked out modern Chinese character frequency list published by MTSU at 2005, and we can see that even there are nearly 10,000 characters, only around a thousand are most commonly used (which contributes over 90%). Figure 19, and Figure 20 show the raw frequency char and accumulative frequency chart in percentile respectfully for Chinese character.

Therefore, based on the knowledge of these, in order to build our Chinese stroke recognizer, we try to find the shapes we want to consider as the primitives to recognize, which could both cover all the intro level Chinese character but also not introduce confusions to students and difficulty to the recognizer. By check the top 100 hundred characters, we break them into strokes, and conclude with 16 most common Chinese strokes that cover all the common characters, and intro level Chinese characters, while eliminating other rare strokes. This brings two benefits: simplifying the recognition and not confusing the students.

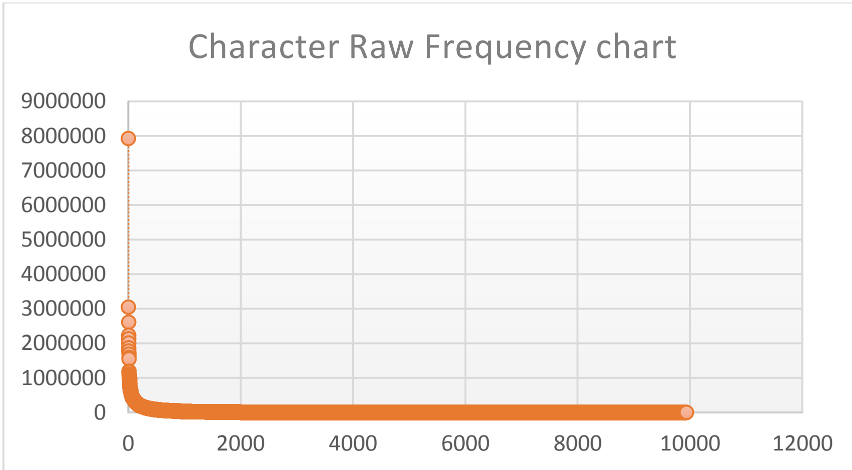


Figure 19 Chinese character raw frequency in 200 million character corpus

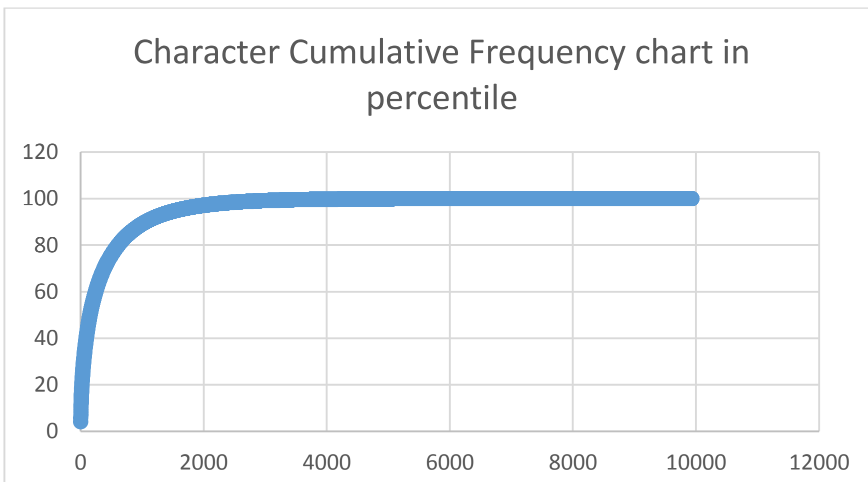


Figure 20 Related cumulative frequency in percentile of Chinese character

Figure 21 is the summary the strokes extracted. On the left side is eight basic strokes, and on the right is eight compound strokes.

Heng	(横)
Shu	(竖)
Pie	(撇)
Na	(捺)
Dian	(点)
Ti	(提)
Wan	(弯)
Gou	(钩)

HengPie	(横撇)
HengZhe	(横折)
HengZheWanGou	(横折弯钩)
HengGou	(横钩)
HengZheTi	(横折提)
ShuZhe	(竖折)
ShuWanGou	(竖弯钩)
ShuZheZheGou	(竖折折钩)

Figure 21 Strokes used in Chinese Calligraphist

The shapes are very different from other sketch system those for architecture, mathematics, electronic circuit design, etc. Those shapes are more likely to be random, mixing with circle, curve, helix, hex, ellipse, lines, arcs, etc. The difference and gap between each other shape is more obvious. Also, for some one stroke complex shape, it is easier to divide them into more primitive shapes and then to segment and combine them later on with the context and shape definition.

On the other hand, Chinese stroke shapes shown as above have different features. Firstly, one stroke shape, even if it is a compound stroke which contains the concept of multiple basic strokes, should be written in one pen movement without any break. When we consult a dictionary, we could look for a character based on the stroke number. If we break one stroke into multi-parts, we also break the idea of the stroke system, therefore it will confuse students, especially non-native western students. As a consequence, when we build a sketch based learning tool with recognition, we should not break stroke into parts, but keep them as a whole unit.

Secondly, writing technique is an important metric to measure whether the shape is written correctly or not. Each stroke has its own desired writing techniques, such as direction, how positive or negative slope is allowed, how concave or convex is the bending, how long is the stroke comparing to its neighboring strokes, etc. All these metrics are used to evaluate the writing technique. However, the most important feedback and evaluation, essentially to encourage students to learn and build confidence in learning writing Chinese is that how the written shape looks like the stroke. Therefore, to provide additional information for guiding students, we could consider all the metrics measurements, but when deciding the correctness, the visually similarity is still the major consideration.

Finally, we find that the similarity between Chinese strokes is potential issue for recognizing as well.

Table 6 Chinese stroke similar groups

Similarity Group	Stroke Symbols		
ShuZhe, ShuWanGou			
Pie, Ti			
Heng, HengGou			
Na, Dian			
Shu, Wan, Gou			
HengZheTi, HengZhe, HengPie			

In Table 6, we summarized some similarity groups of the extracted 16 strokes that will be considered in stroke recognition. The majority similarity happens when some strokes only add a hook at the end of some others, or when some stroke is more bending then another. For example, in the group ShuZhe and ShuWanGou, the later one has a hook at the end but the former one does not; in the group Dian and Na, Dian is short and straight while Na is bending concave and long; and in group Pie and Ti, Ti is line while Pie is curve. One thing to mention is that Ti and Pie are in different writing direction. Pie should be written from upright end to downleft, but Ti should be written from downleft end to upright quickly and shortly.

All these features of Chinese characters and strokes brings some difficulty in recognizing and providing feedbacks, however, fortunately, there is a structure pattern that not only assists us to learn and understand Chinese characters, but also reduces the shapes to be recognized, i.e. only countable Chinese strokes, so that it reduces the complexity and chance of misrecognition. The proposed solution of recognizing techniques is discussed in next section.

3.2 Chinese Stroke Recognition

3.2.1 Choosing Recognizer: Gesture, Templates, or Geometry

As discussed in chapter related works, we mentioned that there are generally three types of recognizers: gesture based recognizer, template based recognizer, and geometry based recognizer. Table 7 summarized the advantages and disadvantages of each for recognizing Chinese stroke shapes. Beside the three, there is a radical based recognizer specifically for Chinese recognition domain proposed by Long-Long Ma. In his method, it solves the issue of segmentation of radicals and builds a radical tree. When a radical is recognized, it searches path along the tree through the radical. However, the false positive and segmentation of strokes is still a problem.

Table 7 Comparison among different types of recognizers for Chinese domain

	Pros	Cons
Gesture Based [rubine, long]	Fast recognizing	Scalability, Similar gestures
Template Based [dollar, hausdorff]	Scalability Order/Direction Free	Sensitive to training data Large number of templates Slow (accuracy – speed tradeoff)
Geometry Based [paleo, sezgin]	Domain independent Training data free	Hard to define complex shapes Improper for Chinese domain
Radical Based [Long-Long Ma]	Recognizing radicals Chinese domain	Segmentation, False positive

From the table above, we want to design a recognizer that takes advantages of each and suppresses the disadvantages. For template recognizers, the tolerance of diverse writing techniques makes the recognition more reliable and versatile. On the other hand, when template recognizer fails – for instance, Na and Dian are similar after we apply pre-processing, the misclassification happens – gesture recognizers can distinguish them with additional information such as stroke length or curviness, etc. Therefore, we proposed a multilayer perceptron recognizer with using the features returned by both gesture and template recognizers. Thus in this way we have the power to combine and determine the usefulness of each recognizer. By doing feature selection, we could analyze how different recognizers are collaborative with each other, and suppress the cons by different trainings.

3.2.2 MultiLayer Perceptron Recognizer

Multilayer perceptron is a classifier that takes a bunch of features as inputs, and then calculate both the weights between inputs and hidden layer units (HLU) and the weights between HLUs and output classes with using gradient error feedbacks. WEKA is an open source tool that implements a well-established version of multilayer perceptron classifier, therefore we integrate WEKA API to build the recognizer so that we do not reinvent the wheel.

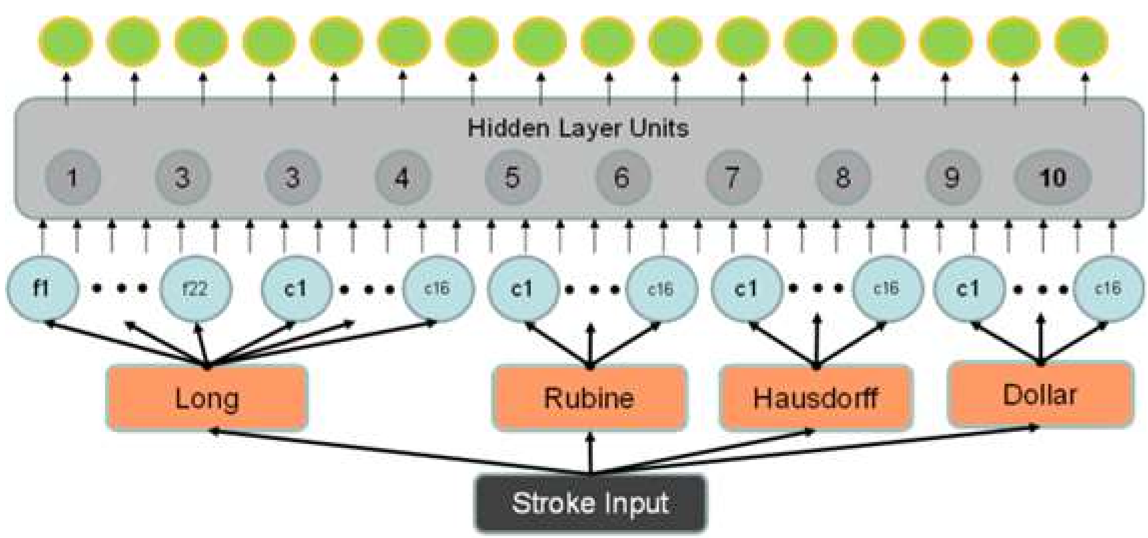


Figure 22 Features and neural networks structure of designed MLP recognizer

Figure 22 shows the proposed Chinese stroke recognition scheme. There are four stages: input, basic recognition, feature calculation, and multilayer perceptron, and output. In input stage, using the application, user draws Chinese strokes free-handily. The strokes are stored as list of points and time information and are sent into four basic recognizers: Long recognizer, Rubine recognizer, Hausdorff recognizer, and One Dollar recognizer in second stage. Follow up is the third – feature calculation stage, in which the output confidence or distance values are normalized. The fourth and fifth stage are computed using WEKA multilayer perceptron classifier API, and output the final confidence of each Chinese stroke class.

We remain the confidence of the recognizing result of four recognizers: two of template, Hausdorff and Dollar, and two of gesture, Rubine, and Long. Each recognizer output the normalized confidence of the 16 shapes, while Long outputs additional 22 features which are those used in Long recognition itself. Altogether these 86 features are the simplest and intuitive ones. Even though there could exist some more Chinese stroke domain relevant features such as bendiness, concaveness or convexness, end-hookness, number of corners, stroke length, etc, we still believe that these intuitive features could make it possible to remain benefits from their advantages. However, we would like to test and design new features that are more relevant and decisive in next version.

Since the absolute confidence values are in different scale, and most of them are influenced by the stroke shape such as length, the only matter numbers are the relative ranking or normalized value. We designed different methods of normalization for the four recognition results based on their value features. For Rubine, and Long, the value number sometimes is very large, and even could be negative, so we applied the standard normalization method:

$$\boldsymbol{range} = \mathbf{max}\left(\boldsymbol{V}_{(\boldsymbol{interpretations})}\right) - \mathbf{min}\left(\boldsymbol{V}_{(\boldsymbol{interpretations})}\right) \qquad \cdots (1)$$

$$\boldsymbol{normalized\_confidence_{I}} = \left(\boldsymbol{V_I} - \mathbf{min}\left(\boldsymbol{V}_{(\boldsymbol{Interpretations})}\right)\right) / \boldsymbol{range} \qquad \cdots (2)$$

For hausdorff confidences, we designed another normalization method. Since hausdorff metric measures the distance between the shape and templates, it can be even zero when they are identical in looking. The normalization is to inverse the distance with comparing the minimal value:

$$\boldsymbol{distance}_{(\boldsymbol{interpretation})} = \mathbf{min}\left(\boldsymbol{Distance}\left(\boldsymbol{Templates}_{(\boldsymbol{interpretation})}\right)\right) \qquad \cdots (3)$$

$$\boldsymbol{normalizer} = \mathbf{1} + \mathbf{min}\left(\boldsymbol{distance}_{(\boldsymbol{interpretation})}\right) \qquad \cdots (4)$$

$$\boldsymbol{normalized\_confidence}_{(\boldsymbol{I})} = \left(\mathbf{1} + \boldsymbol{distance}_{(\boldsymbol{I})}\right) / \boldsymbol{normalizer} \qquad \cdots (5)$$

For Dollar, since it is using 0-1 scaled scoring confidence itself, thus, it remains the same:

$$\boldsymbol{normalized\_confidence}_{(\boldsymbol{I})} = \mathbf{1} - \boldsymbol{distance}_{(\boldsymbol{I})} / \left(\mathbf{1/2} \times \sqrt{\boldsymbol{size}^{\mathbf{2}} + \boldsymbol{size}^{\mathbf{2}}}\right) \cdots (6)$$

where size is 40, which is explained in preprocessing section.

Obviously not all these features are essential. Most features are overlapping telling the same thing redundantly, i.e. some of them could be removed to accelerate and simplify recognition. This is feature selection. In our method, we designed to do feature selection with 10 fold cross validation, and to see what features are selected the most times and what are not selected at all. Weka provides attributes selection experiments. For attribute evaluator, we chose classifierSubsetEval and set the classifier as multilayer perceptron, and we also chose GeneticSearch as the search method with default argument setups.

In order to overcome the disadvantages of the gesture recognizer, we try to mess-up the training data so that it seems more robust. The strokes contains different writing techniques. In order to overcome the disadvantage of the template recognizer, we try to reduce the template number so that less matching, and we want to see how well it works.

3.2.3 Preprocessing

The preprocessing steps are used in hausdorff and dollar recognition. Three steps are taken in place: translate, resampling, and resizing. The first step is to translate the upleft corner of the bounding box to the origin. That is $offset = (-\min(X), -\min(Y))$, and for each point $point = point + offset$. The second step is to resample the shape to a fixed number of points. There are several options, 32, 40, and 64. In Ayden's thesis, he compared 64, and 32, and concluded with 32 won better recognition for his kid's drawing domain. We adopt the Dollar's 40 by default. Therefore, in the calculation of Dollar's score, the size is set as 40. The resampling algorithm is shown as Figure 23: calculate the unit length, and search along the point path. Accumulate the length until it is longer than the unit length, get one resampling point, and restart searching from this point, until find all the resampling points. If the gap between two points is too long, it would recursively break into unit length.

```
RESAMPLE(points, n)
    I ← PATH-LENGTH(points) / (n − 1)
    D ← 0
    newPoints ← points_0
    foreach point p_i for i ≥ 1 in points do
      d ← DISTANCE(p_{i-1}, p_i)
      if (D + d) ≥ I then
        q_x ← p_{i-1_x} + ((I − D) / d) × (p_{i_x} − p_{i-1_x})
        q_y ← p_{i-1_y} + ((I − D) / d) × (p_{i_y} − p_{i-1_y})
        APPEND(newPoints, q)
        INSERT(points, i, q)   // q will be the next p_i
        D ← 0
      else D ← D + d
    return newPoints
PATH-LENGTH(A)
    d ← 0
    for i from 1 to |A| step 1 do
      d ← d + DISTANCE(A_{i-1}, A_i)
    return d
```

Figure 23 Reampling algorithm adopted

The final step is to resize. In order to maintain the width and height as the same proportion as the origin drawing, we resize the x, and y with the same factor. Because in resampling step, we adopted 40 resampling points, therefore the maximum x, and y should be 40 so that any horizontal and vertical line can be resampled properly. So we choose the larger one between width and height, and then divided by 40. Finally, apply it to every point. It is shown as the following equations, and Figure 24 is the examples of the some strokes before and after preprocessing.

$$i = argmax(width,\ \ height)$$

$$resizeFacter = 40.0\ /\ i$$

$$for\ each\ point: (x, y) = (x, y) \times resizeFacter$$

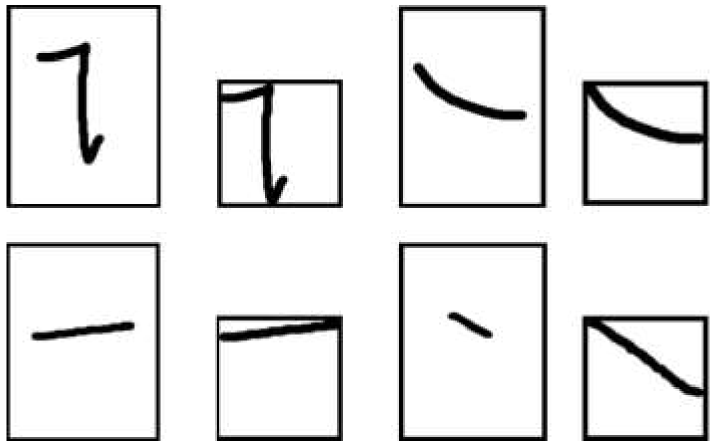
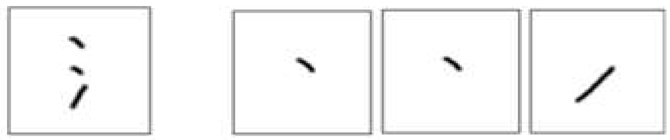


Figure 24 Comparison of stroke shapes between before and after preprocessing

3.3 Chinese Character Recognition

After strokes are recognized, the next step is to combine and segment them into radicals and characters. We are using the concept of LADDER to implement our character recognizer. The ladder recognizer starts checking the spatial relationship with the relative size of the strokes on canvas. When a stroke recognition result pops out, it starts building candidates characters. Once satisfying all constraints of any character, it confirms the one and removes others sharing parts of strokes with this character, and start checking writing techniques. For defining ladder domain shape, including components, alias, spatial constraints, we build character domain shape defining XML files as Figure 25 through 27.

In Figure 25, we see that for each shape, there is an assigned name and type. These two together identify a shape. The type could be character, radical, stroke, and observation. The component is a list of shapes that make the shape. The attribute alias is used to distinguish same subshapes. Once all the subshapes are filled in, the constraints checking starts. Every constraints is a rule of the shape which takes two components as parameters, and define their spatial relationship. If all checks returns true, the shape is recognized.




```xml
▼<Hypothesis name="SanDianShui" type="Radical">
  ▼<components>
      <component alias="dian1" name="Dian" type="Stroke"/>
      <component alias="dian2" name="Dian" type="Stroke"/>
      <component alias="ti" name="Ti" type="Stroke"/>
    </components>
  ▼<constraints>
      <constraint check="Above" para1="dian1" para1Spec="Bottom" para2="dian2" para2Spec="Top"/>
      <constraint check="Above" para1="dian2" para1Spec="Bottom" para2="ti" para2Spec="Top"/>
      <constraint around="5" check="SameX" para1="dian1" para1Spec="Left" para2="dian2" para2Spec="Left"/>
      <constraint around="5" check="SameX" para1="dian1" para1Spec="Right" para2="dian2" para2Spec="Right"/>
      <constraint around="10" check="SameX" para1="dian1" para1Spec="Left" para2="ti" para2Spec="Left"/>
      <constraint around="10" check="SameX" para1="dian1" para1Spec="Right" para2="ti" para2Spec="Right"/>
    </constraints>
```


Figure 25 LADDER definition of radical "Water" in XML view


```xml
▼<Hypothesis name="You" type="Character">
  ▼<components>
      <component alias="hengpie" name="HengPie" type="Stroke"/>
      <component alias="na" name="Na" type="Stroke"/>
    </components>
  ▼<constraints>
      <constraint check="Intersect" para1="hengpie" para2="na"/>
      <constraint check="Above" para1="hengpie" para1Spec="Top" para2="na" para2Spec="Top"/>
      <constraint check="Right" para1="na" para1Spec="Right" para2="hengpie" para2Spec="Right"/>
      <constraint check="Left" para1="hengpie" para1Spec="Left" para2="na" para2Spec="Left"/>
    </constraints>
```


Figure 26 LADDER definition of character "Again" in XML view

```xml
▼<Hypothesis name="Han" type="Character">
  ▼<components>
     <component alias="sandianshui" name="SanDianShui" type="Radical"/>
     <component alias="you" name="You" type="Character"/>
   </components>
  ▼<constraints>
     <constraint check="Left" para1="sandianshui" para1Spec="Right" para2="you" para2Spec="Left"/>
     <constraint check="Above" para1="sandianshui" para1Spec="Top" para2="you" para2Spec="Top"/>
     <constraint check="Below" para1="sandianshui" para1Spec="Bottom" para2="you" para2Spec="Bottom"/>
```

Figure 27 LADDER definition of character “Chinese” in XML view

Figure 27 is the ladder definition of character “汉”, or “Han”, which means Han dynasty, Chinese, or Chinese ethnic group. It has components of two radicals: “氵”, “SanDianShui”, and “又”, “You”, which are shown in Figure 25, and Figure 26 respectively. This hierarchical, recursive XML file structure makes it simple and clean for each shape. And it accords with the top-down Chinese character structure feature while corresponding the bottom-up recognition design.

3.4 Error Detection and Feedbacks

It is similar, or rather the same, with ladder constraints. This constraint checks only happens after the recognition of a shape (character, radical, stroke) is confirmed, i.e. every check in constraints list has passed. This feedback is not affecting the recognition, but will provide the information of the feedbackString in feedback panel in the user interface. This feedbacks are generally about the writing techniques such as stroke orders. Figure 28 is an example of the feedback of the radical "氵", "SanDianShui".

```xml
▼<feedbacks>
   <feedback check="Before" feedbackString="Dian at top should be written before Dian at middle" para1="dian1" para2="dian2"/>
   <feedback check="Before" feedbackString="Ti at bottom should be written at last" para1="dian2" para2="ti"/>
 </feedbacks>
```

Figure 28 LADDER feedback design of radical "Water" in XML view

4. IMPLEMENTATION

4.1 System Architecture

The application is focusing on teaching students to write Chinese characters. It provides lessons which is a set of problems to practice, and the canvas for writing, as well as feedback panels for reviewing. It has several modules: user interface, recognition, feedback, domain parsing, and lesson information, shown in Figure 29.

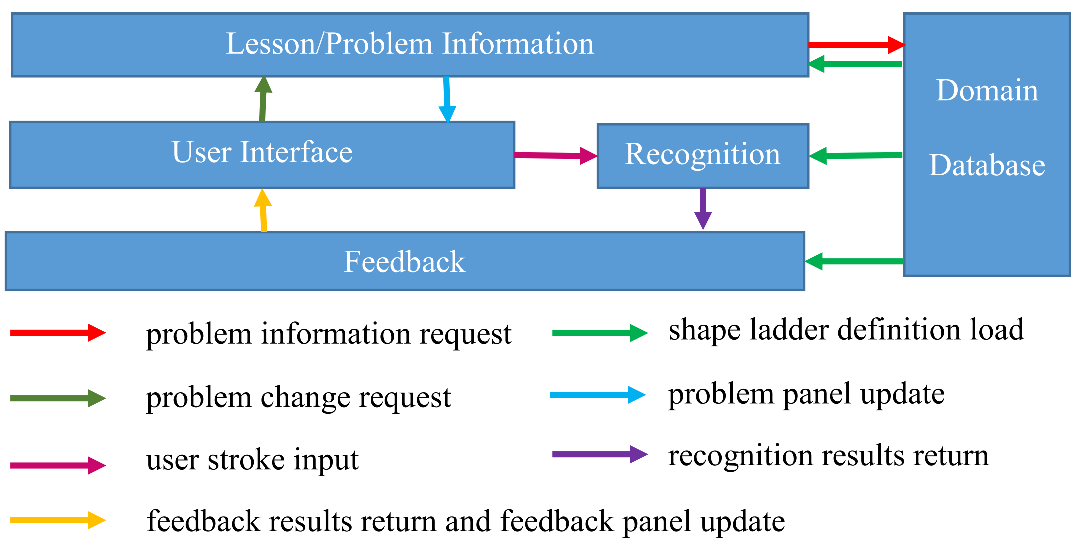


Figure 29 Interface data flow and control flow

When a student selects lesson, it sends request to get the problems and their related shape ladder definition. From the domain database, XML files, it loads lesson problem list with description, and then update the user interface to show the new problems. Meanwhile, the recognizer updates and load the shape list recursively so that it ignores irrelevant ones that are not used in the lesson.

When a student write strokes on the canvas, the low level primitive multilayer perceptron recognizer output the interpretation result, which is the observation shape in high level recognition. Then high level character ladder recognizer take the observation shape to build stroke shape, and candidates radical and character shapes with this stroke shape. The character shape is listening and check whether it fills as a component whenever a new stroke is written. If all the component is filled and the constraints checking is passed, then the shape is recognized and remove other candidates sharing the same components. The recognition results will return to feedback module. The feedback module checks whether it satisfying all the feedback constraints, and returns warning or confirmation notes back to the user interface, by updating and showing on the feedback panel.

4.2 Stroke and Shape Data Structures

The stroke data structure is used to store the sketching points and low level recognition results. Figure 30 is the UML structure of class Stroke, Interpretation, Points, etc. Besides, the stroke data structure also maintains the information of bounding box. This will be used in hierarchical shape recognition, so that the recognizer can check the spatial relationship between strokes, or shapes.

On the other hand, shape data structure is the unit used in ladder recognition. Each one is assigned with a XML definition file. In constructing method, it reads the XML and builds the constraints and feedbacks. Also, in the XML, it defines the list of components.

The BoundingBox is built and updated while strokes are added. The check() method of DomainShape calls the check() method of all constraints. They are defined in the XML file as well. Figure 31 is the UML graph of the classes relevant to the DomainShape class.

Figure 30 The UML of class stroke

4.3 Feedback Contents

As described in last chapter, feedback is similar concept as constraint. They use the same checking methods but the difference is that feedback checking is not included in recognition. The reason doing this is that we want the recognition is checking the visually correctness while the feedback is checking the writing technique correctness, which is stated in Paul's thesis. When learning writing Chinese, the students are trying their best to make the writing close to the shape shown in the textbook and blackboard. They may make mistakes in some stroke orders or not do well in stroke beautifications, but this does not make the overall writing wrong. If the shape the student writes and the textbook shows are visually almost the same but the recognition and feedback says wrong, this is definitely very discouraging and make students upset and unconfident either. We want to encourage the students' passion of learning, but not dampening it. Therefore, the result is that we tell students the characters are recognized, which will confirm and comfort them, while we tell them as well how well are their writing.

We designed three types of feedbacks: general feedbacks, stroke feedbacks, and character feedbacks. In general feedbacks, we tell the students 1> what stroke you are writing; 2> what radicals and characters are already written on canvas, i.e. recognized, and 3> whether the stroke is desired one. First two feedbacks is displayed verbally on the feedback panel which lays on the right of the sketch panel, and the third one will show as the background image change of the desired stroke in the writing instruction panel.

Figure 31 The UML of class DomainShape

In stroke feedbacks, we tell students whether stroke direction is written correctly, and several point positioning relationships such as if Heng is written as a negative slope line. For character feedbacks we only check the stroke order correctness. Table 8 is the summary.

Table 8 Contents to check in different types of feedbacks

General Feedback	Stroke Feedback	Character Feedback
Recognized strokes	Start End Point Position	Stroke Order
Recognized radical/characters	Checks: Above, Below, Left, Right	Checks: Before
Match of written and wanted strokes		

In practice session, we have the information of the stroke orders and desired stroke information. So once a student write a stroke, we could compare it with the recognized stroke. If they doesn't match, besides saying they are not matching and ask for the student rewriting, it is also plausibly to tell how and why those two strokes are different with the recognizing features. For example, if Heng is wanted, however a student write a stroke which recognized as Ti, from the slope we can provide feedback that the positive slope it too much to be a Heng so that it becomes a Ti. This is an example, we can discover and develop more feedback features.

4.4 Interface Design

We designed an easy and clean interface in order to make students focusing on writing. Therefore the majority of the GUI is the sketching area, and other functional parts are located surrounding the sketching panel and shall not distract students' attention too much. Figure 32 is the overview of the GUI. Overall it contains different panels: Problem Description Panel, Character Guides Panel, Sketching Panel, Controlling Panel, and Feedback Panel which are shown in Figure 33 to Figure 37 respectively.

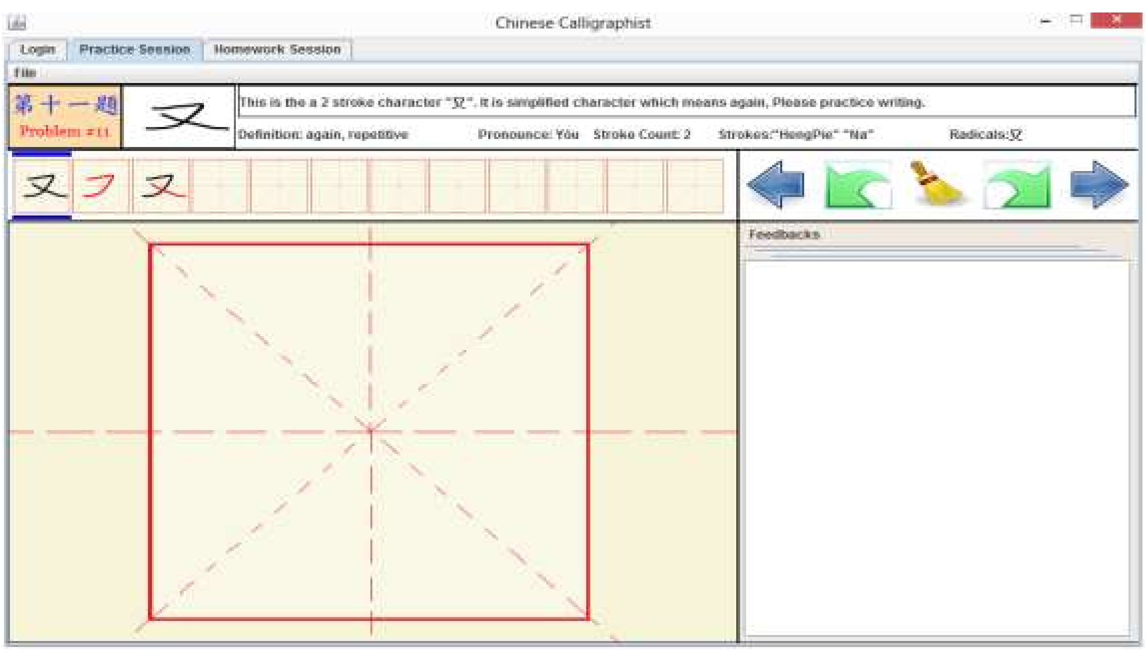


Figure 32 Overview of the Chinese Calligraphist GUI

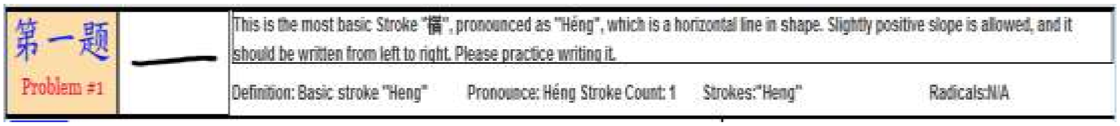


Figure 33 Problem information panel

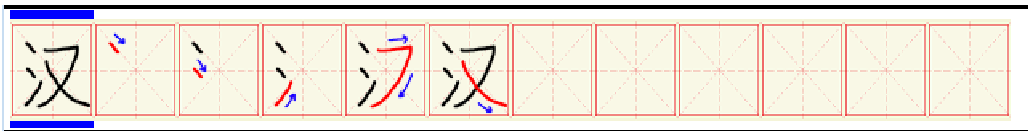


Figure 34 Character stroke guiding and feedback panel

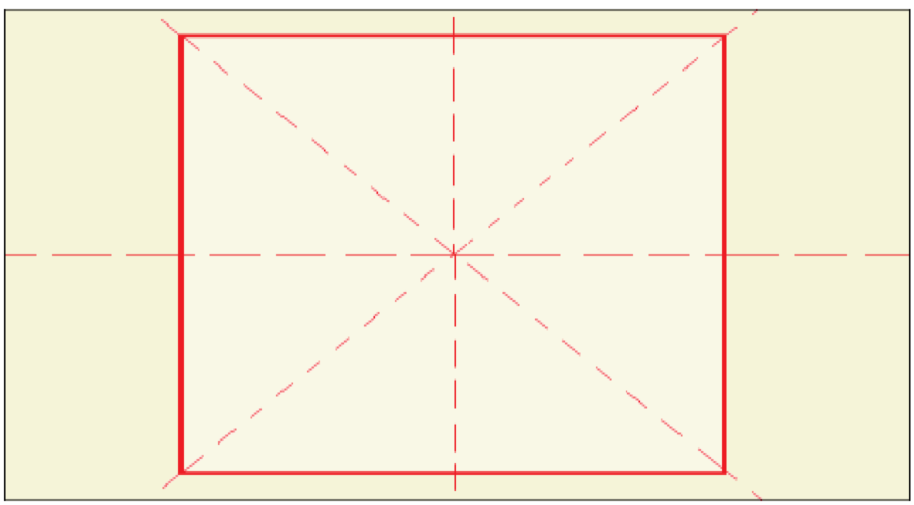


Figure 35 Input and sketch panel

Figure 36 Interactive control buttons

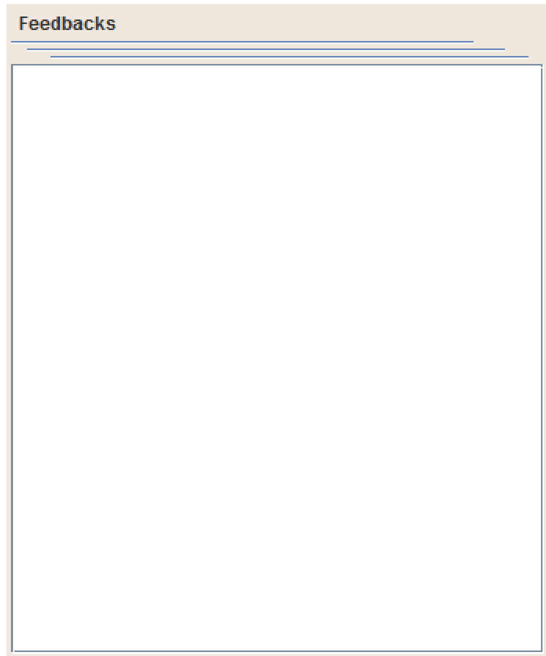


Figure 37 Text feedback panel

The Problem Description Panel is in three columned structure. The first is the counting number; the second is the character shape; and the third is the text of problem. It will explain the definition of the character, stroke lists, radical lists, and pronunciation of the character.

The Character Guide Panel is the stroke splitting graph list of the character. Each graph is responding to the correctness of each stroke written on the sketching panel. For example, in Figure 38, three strokes are written correctly hence all three shapes are turning green in background. However, in Figure 39, the third stroke is written wrongly, therefore the background of the third stroke becomes red.

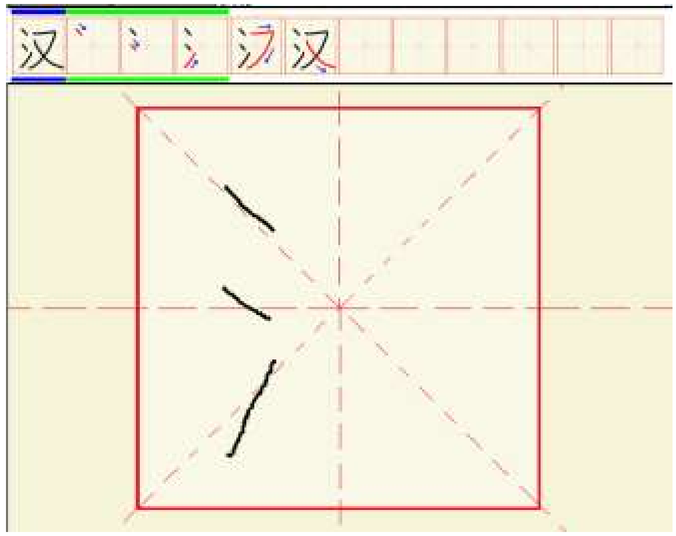


Figure 38 Confirming correct writing feedback

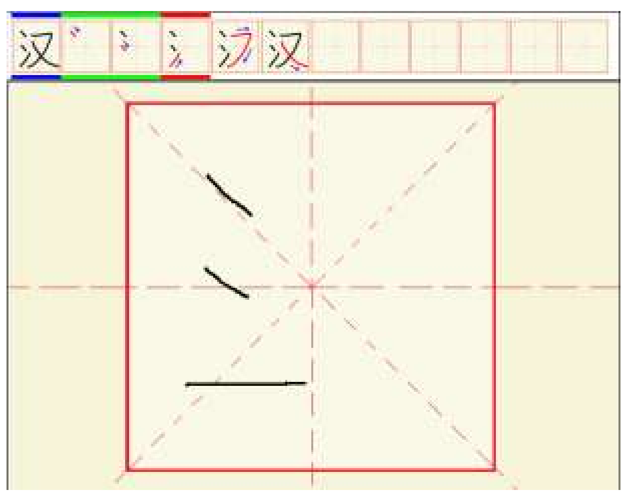


Figure 39 Warning incorrect writing feedback

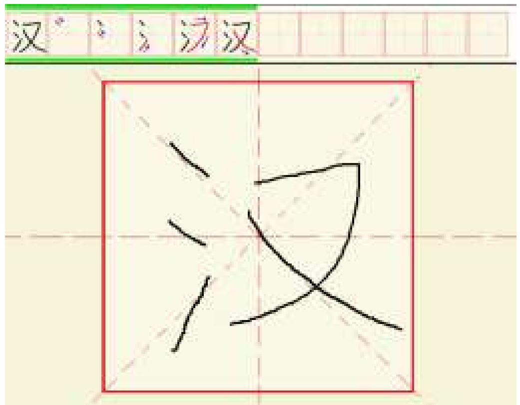


Figure 40 Dispalying complete and correct writing feedback

When the character is completed, the background of the first shape, which is blue before, will also turn green, shown as Figure 40. The feedback panel will show the stroke, radical results, as well as the writing technique correctness, shown in Figure 41.

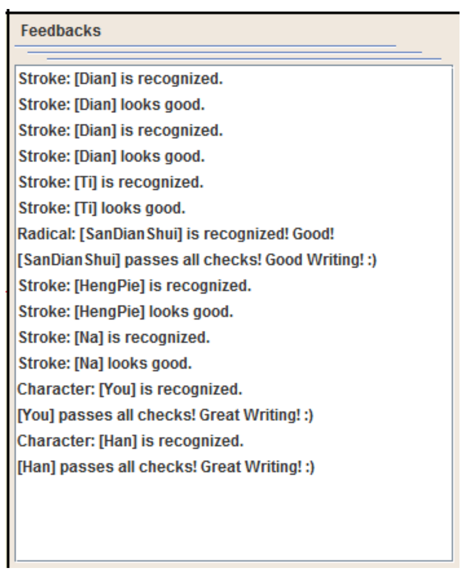


Figure 41 Text feedbacks about the recognized strokes, radicals, and characters

5. EXPERIMENTS AND EVALUATION

5.1 Data Collection

5.1.1 Sousa

To collect data for training the recognizer, a study in Sousa is created. Sousa is a web tool for studying hand drawing from TAMU SRL. In the study, it differentiate the user by the Chinese writing skills. The native's data is used for training, so that reduces the chance of bad instances. On the contrary the data from non-native students are used for recognition testing. There are 2 native Chinese students who are regarded as good writers writing the templates. Each of whom writes 160 strokes. 10 templates for each of the 16 strokes (8 basic strokes, and 8 compound strokes). They are written with the Wacom bamboo device, shown as Figure 42.

Figure 42 Wacom bamboo device used in collecting training and testing data

For testing, we have two different data collections:

1> Collecting testing data for low level recognition: we have 1 native student and 1 non-native student writing 10 times for each 16 strokes, which are the identical same shapes as in collecting training data. In total these 320 strokes are used to test the performance of the stroke recognition. And all these data are written in Chinese Calligraphist tool.

2> Collecting data for user study: we have 4 non-native students taking lesson 0 and lesson 1 which contains 12 strokes, 2 radicals, and 14 characters. They are divided in two groups, and one group is writing on paper with pen, the other is writing through Sousa. Through sousa, we have 65 character data from each students.

5.1.2 Choosing Input Method

Generally, the input method for electronic writing are divided in three categories: mouse, finger, and stylus. Comparing with fingers, and mouse, both native and non-native writers are more comfortable with the pen. When using mouse, it is too shaky to write strokes smoothly. This introduces lots of noise in both making templates and doing recognition. When using fingers on the pads, the fat finger effect affects the writing experience and the recognition. Therefore, we select using stylus/E-pen as input, since it is also the closest writing experience with the standard writing.

5.2 Evaluation Design

After collecting data, we train the recognizer, and test the recognition algorithm. At first, using the training data to train Rubine, Long and get the weight matrix. And then evaluate the stroke recognition with the testing data collected in (1). After that, with these recognition confidence results for all four recognizers, as well as the 22 features from Long, train the multilayer perceptron recognizer. Evaluating the multilayer perceptron recognition performance, we use weka to test the 10 fold cross validation. All these evaluation results are recorded with confusion matrix and f-measure.

Except for evaluating the difference of recognition algorithm, the effect of number of templates is also studied. We tested and compared the performance of multilayer perceptron recognizer under 20, 10, 5, 2, and 1 templates separately.

The effect of feature set is another evaluation in our study. Initially there are 86 features totally. However, many of them are overlappingly and redundantly telling same thing, hence it reduces the information gain and increases the overhead. Feature selection is to evaluate the importance of each feature, and find the optimal solution for the feature combinations. Weka is still being used as the tool. For each template number, we also run the GeneticSearch to find the best feature subset, and evaluate the f-measure as well. In addition, we designed another two tests: (1) doing 10 fold cross validation feature selection, we select the subset features that are selected more than 90% times. (9 out of 10), and apply them with different template number situation to see how it performs; (2) union the subset all together for each templates, rerun feature selection with the feature union to find the final optimal solutions, and select the best template number. The goal of all the tests is to find the best subset of current feature set so that we can both reduce the number of features, which simplifies the building classifier and recognition, and reduce the template matching during recognition, which reduces the complexity of recognition as well.

Evaluating character recognizer, we use the testing data collected from user study. This is done and recorded manually since the stroke is input using Sousa and replay it back with Chinese Calligraphist tool. We use the data of 130 examples about 13 characters, (5 example per student, per character, and 2 students) to record the recognition result.

Evaluating the quality of feedbacks, we ask the students for their idea after the user study. They learn writing Chinese characters using pen or sousa. And get feedback respectively. After that, they are invited to use Chinese Calligraphist, and to see how the instant and interactive feedbacks are given. And we asked several questions about their using experience, which is discussed in the following section.

5.3 User Study Design

As mentioned before, the users are split into two groups to learn writing Chinese. One uses paper and the other uses our interface. For those who use Chinese Calligraphist, we created 2 lessons on sousa for both two students. Each lesson have practice study and quiz study. In practice study, the goal is to help them get familiarized with Chinese writing, stylus using, and Chinese stroke, character knowledge. They can write as many times as they want to. In quiz study, there are several writing tests for some characters. Each character test requires writing 5 times. To provide feedbacks, these writing records are replayed in our interface, and the feedbacks of the writing offered by the interface is provided to the students. For those who use paper and pen, the traditional feedbacks about the visually correctness is provided. There is no way to check the writing technique correctness afterwards.

After they get feedbacks of the lessons. They are all invited to use the interface to experience the instant and interactive feedback. Meanwhile, the ability about writing skills, such as how many incorrect writing occurred, of both groups will be recorded, compared to each other. Afterwards, both group give feedbacks of their feeling about the study via interview and/or survey questions. The questions will focus on the friendliness, convenience, feedback quality, shown as Table 9. Future study can be applied to the Chinese courses from Department of International Studies at Texas A&M University.

Table 9 Survey questions for feedbacks of the interface after user study

1	Is the UI clean and natural as using pen and paper? (1 – distracting to 5 – natural as paper mode)
2	Is the problem definition clear and enough? (1 – confusing to 5 – perfect)
3	Is the stroke recognition results convincing? (1 – mostly wrong to 5 – persuasive and instructive)
4	Is the recognition timely? (1 – instant to 5 – very laggy)
5	Are the feedbacks correct and clear? (1 – unclear to 5 precise and straightforward)
6	Is the image background feedback marked? (1 – hard to notice to 5 – striking)
7	Is the text feedback marked? (1 – hard to notice to 5 – striking)

5.4 Results

5.4.1 Stroke Recognition Accuracy

Table 10 shows the confusion matrix of the recognition testing results of using Rubine recognizer. Generally it works maintains high accuracy, but for stroke Na, Wan, Gou, ShuWanGou, and ShuZheZheGou is not distinguishing well. We can also see that

Wan and Gou is a confusing pair. In shape they are similar themselves. Gou is nearly Wan with a hook up at the end. The hook, comparing to the main part of the stroke, is short and small. The difference of the angle, length features might be very small. And that's why many Gou are misrecognized as Wan. Moreover, Shu is also a similar shape with these two. Shu is more a line than a curve, but these three strokes are in one category. Thus this verifies the hypothesis and statement about the similarities of Chinese strokes.

Table 10 Confusion matrix for Rubine recognition result

Actual Stroke		H	S	P	N	D	T		W	G	HP	HZ	HZWG	HG	HZT	SZ	SWG	SZZG
RecognizedAs Stroke		横	竖	撇	捺	点	提		弯	钩	横 撇	横 折	横折 弯钩	横 钩	横 折 提	竖 折	竖 弯 钩	竖折 折钩
H	横	19			2													
S	竖		20						4	4								
P	撇			19														
N	捺				16													
D	点					19												
T	提						20											
W	弯								15	7								
G	钩			1					1	9	1							
HP	横撇										19							7
HZ	横折				1							19						
HZWG	横折 弯钩												19					
HG	横钩	1												20				
HZT	横折 提														20		5	
SZ	竖折				1	1						1				20		
SWG	竖弯 钩												1				15	
SZZG	竖折 折钩																	13

Table 11 is showing the confusion matrix for Long's testing results. It is pretty the same as Rubine, for example Wan, Gou, and Shu. But it overall is superior.

Table 11 Confusion matrix for Long recognition result

Actual Stroke		H	S	P	N	D	T	W	G	HP	HZ	HZWG	HG	HZT	SZ	SWG	SZZG
RecognizedAs Stroke		横	竖	撇	捺	点	提	弯	钩	横撇	横折	横折弯钩	横钩	横折提	竖折	竖弯钩	竖折折钩
H	横	20			2												
S	竖		20					4	3								
P	撇			20													
N	捺				18	1				1							
D	点					19											
T	提						20										
W	弯							16	5								
G	钩								10	1	3						
HP	横撇									17							2
HZ	横折										16						
HZWG	横折弯钩											19					
HG	横钩									1			20				
HZT	横折提								2					20		1	
SZ	竖折										1				19	2	
SWG	竖弯钩											1			1	17	
SZZG	竖折折钩																18

The recognition confusion matrix for Hausdorff and One Dollar algorithm are shown in Table 12, and Table 13 respectively. These two exhibit way much worse accuracy than previous two feature based recognition. Template based recognizer relies on the perfect visually matching so much.

Table 12 Confusion matrix for Hausdorff recognition result

Actual Stroke		H	S	P	N	D	T	W	G	HP	HZ	HZWG	HG	HZT	SZ	SWG	SZZG
RecognizedAs Stroke		横	竖	撇	捺	点	提	弯	钩	横撇	横折	横折弯钩	横钩	横折提	竖折	竖弯钩	竖折折钩
H	横	20			2								1				
S	竖		20					1	6								
P	撇			15			12										
N	捺				18	6											
D	点					12											
T	提			5			0		1	1							
W	弯						8	19	6					1			
G	钩								6								
HP	横撇									16	1						
HZ	横折										19						
HZWG	横折弯钩									3		20					
HG	横钩												19				
HZT	横折提					2			1					19		4	9
SZ	竖折														19		
SWG	竖弯钩														1	16	
SZZG	竖折折钩																11

Table 13 Confusion matrix for OneDollar recognition result

Actual Stroke		H	S	P	N	D	T	W	G	HP	HZ	HZWG	HG	HZT	SZ	SWG	SZZG
RecognizedAs Stroke		横	竖	撇	捺	点	提	弯	钩	横撇	横折	横折弯钩	横钩	横折提	竖折	竖弯钩	竖折折钩
H	横	16			1		1										
S	竖		19			4		1	3								
P	撇		1	20				2	7					1			
N	捺	1			17	5											
D	点	3			2	11							1				
T	提						19										
W	弯							17									1
G	钩								8								4
HP	横撇									20	10		1				
HZ	横折										10		7				
HZWG	横折弯钩											19					
HG	横钩												13				
HZT	横折提								2					19			
SZ	竖折														19		
SWG	竖弯钩											1			1	20	
SZZG	竖折折钩																15

The best scenario for template based recognition is that either the shapes are not alike to each other at all, or each shape doesn't have many variations. However, different writing habits introduce many stroke variations. Furthermore, many Chinese strokes looks alike to each other, like Wan, Gou, and Shu mentioned above, therefore, the template based recognition, i.e. Hasudorff and One Dollar, is relatively poor.

However, we can still learn from the results of the two template based recognition. For those shapes are not recognized well using Rubine or Long, such as ShuWanGou, One Dollar recognizer is fully recognized correctly. For those shapes alike visually, such as Dian and Na, Ti, and Pie, HengZhe, HengPie, and HengGou, the feature of writing techniques gives extra information. Therefore, this confirms the assumption that different recognizers are useful and discriminative in different aspects. If they output the same recognizing result, it strengthen the confidence both from visually and writing technically. If the predicts with conflicts, then interesting situation happens. Either similar shape with different writing technique, or similar writing technique but different visual shape are competing. Abandoning either result is not wise. Find the way to combine them and take advantages of both is what we proposed: multilayer perceptron recognition.

Table 14 is the confusion matrix of the recognition testing using multilayer perceptron recognizer. It drastically improve the recognition accuracy. Only 7 out of 320 strokes are misrecognized. Interestingly, Wan and Gou, as well as Dian, and Na, or HengZhe and HengPie, are also no longer any problems. The hybrid recognition, which also take advantages of neural networks, is actually a non-linear classifier. High dimension separates different classes into different areas, and the classifier partition them into zones.

Table 14 Confusion matrix for MultiLayer perceptron recognition result

Actual Stroke		H	S	P	N	D	T	W	G	HP	HZ	HZWG	HG	HZT	SZ	SWG	SZZG
RecognizedAs Stroke		横	竖	撇	捺	点	提	弯	钩	横撇	横折	横折弯钩	横钩	横折提	竖折	竖弯钩	竖折折钩
H	横	20															
S	竖		20														
P	撇			20													
N	捺				19										1		
D	点					20											
T	提						20										
W	弯							20	1								
G	钩								19								
HP	横撇									18			2				
HZ	横折										20						
HZWG	横折弯钩											20					
HG	横钩				1					2			18				
HZT	横折提													20			
SZ	竖折														19		
SWG	竖弯钩															20	
SZZG	竖折折钩																20

In summary, Figure 43 represents f-measure values for each recognition scheme. Feature based recognizer generally is more proper in recognizing Chinese strokes domain than template matching recognizer. Rubine achieves f-measure at 88.125%, and Long can achieve f-measure at 90.3125%. Hausdorff is the poorest one in our test, which only recognized correctly in 77.8125%. One Dollar recognizer reaches f-measure at 81.875%. Superior to all these, our proposed recognition scheme, multilayer perceptron recognizer, recognizes incredibly correctly with f-measure at 97.8125%. It improves from other four schemes on average by 15.7% ((97.8125 – 84.53125) / 84.53125). In next selection, we will discuss about further study on improve our proposed scheme more.

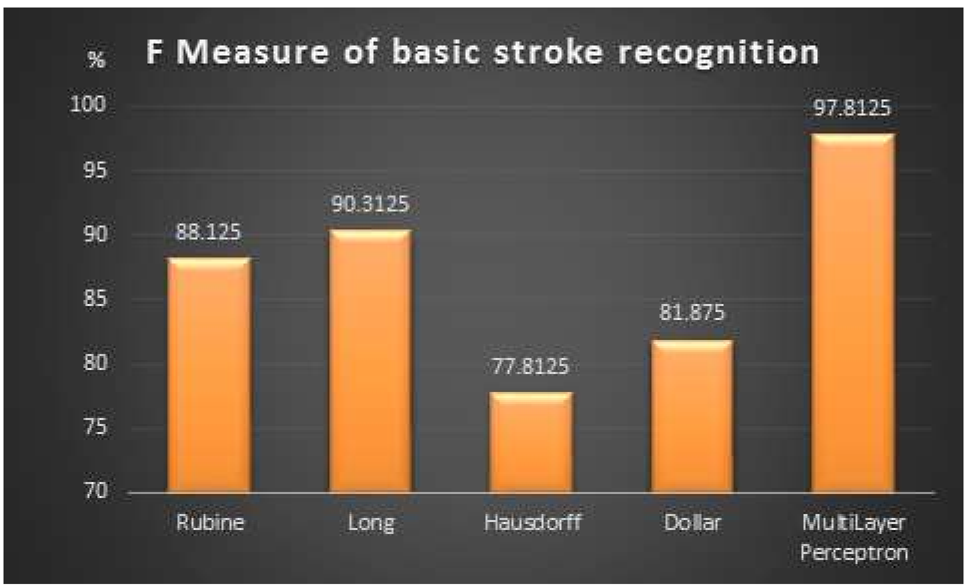


Figure 43 Comparison of f-measures among different recognition schemes

5.4.2 Character Recognition Result Analysis

With the 130 shapes collected by sousa during user study, we observed the performance of recognizing characters. We loaded and replayed the writing behavior of the students. In general, the shapes are look good. However, 84 out of the 130 runs are displaying that the desired characters are recognized.

Strokes are almost recognized correctly. The only typical stroke misrecognition that makes it not recognizing higher level characters successfully is shown as below. The stroke lassoed in green is meant to be Na, however, it is recognized as Dian. The difference between Na and Dian is that Na is more curving, and longer. Look at the two strokes in blue lasso, which is Dian and also recognized correctly as Dian. They look almost the same as the stroke in green lasso. Therefore, it is not wrong recognition, but a good feedback for students.

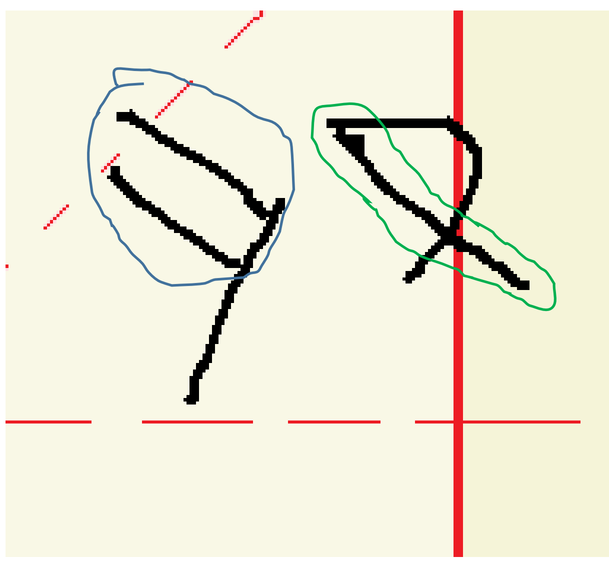


Figure 44 Example of unrecognized characters in user study

The other main reason for not output the desired character recognition is that even though the shape is visually correct generally speaking, in details it fails the pass all the constraint checks in ladder domain definition. For example in Figure 44, in the character 叉 (or radical) on the right, the end of first stroke should be longer and reach to the left to make the character more balancedly spaced. But the writing is too leaning to one side. This helps correct the students' bad writing habits and techniques is important, even it is also discouraging somehow.

5.4.4 Feedback Result Analysis

The feedbacks of characters are about stroke orders. All of the recognized characters provides correct and relevant feedbacks about the stroke order. If it is written correctly, visually and technically, it displays confirming message to encourage students as well.

From the user feedback, they are all happy with the GUI. One user mentioned that overall the interface is clean and straightforward; he liked the problem description panel, and undo/redo button; about recognition result, overall is correct and instant. Image feedback is good, but the text feedback panel is not easy to catch his attention. He suggested that for different types of recognition – Stroke, Radical, and Character – it should use different colors to display to recognition notes. Currently every feedback message is shown in black so that even one new feedback is added, it is not obvious to notice that. Furthermore, for different recognition feedbacks – positive, warning, or negative – it should use different colors, as well, perhaps the background color. In this way, his will pay more attention to the area.

Another user mentioned that at first it is hard to adapt to thinking of satisfying all the visual constraint checks, for the first time use. It writes better after using it more.

5.5 Feature Selection

Using Weka to select features, we choose to use ClassifierSubsetEval as our attribute evaluator. Set the classifier multilayer perceptron with same configurations: learning rate at 0.3, momentum at 0.2, and hidden layer units at (#features + #class) / 2. For subset searching algorithm, we adopted GeneticSearch with default configuration: crossover probability at 0.6, mutation probability at 0.033, maximum generation at 20, and population size at 20.

In our experience (1), select features based on full training set, 14 features are selected which are f6, f12, 15, f16, f17, f21, rubine_ShuZhe, rubine_Heng, rubine_ShuWanGou, rubine_Shu, rubine_Na, long_Heng, dollar_Gou, and dollar_HengPie. F6 tells about the relationship between start and end points, f12 through f21 is more about the shape information: is it thin or square-like shape; or is it long or short stroke, etc. Other confidence values are selected mainly for distinguish the similar stroke shapes and recognition conflicts between recognizers. For example both long_Heng and rubine_Heng are selected. Rerunning the recognition with the selected features, we surprisingly get the same performance, which achieves the f-measure at 97.8125%. This is an exciting news since 84% of the 86 features are omitted while the remaining features still works well as before. It removes huge number of hidden layer units and the number of connecting weights in the neural networks.

Another interesting result is that no hausdorff features are selected. Though it keeps two OneDollar features, the majority are still mathematical features, rather than template features. That makes us confident of reducing the number of templates, so that we are able to improve the efficiency of recognition further.

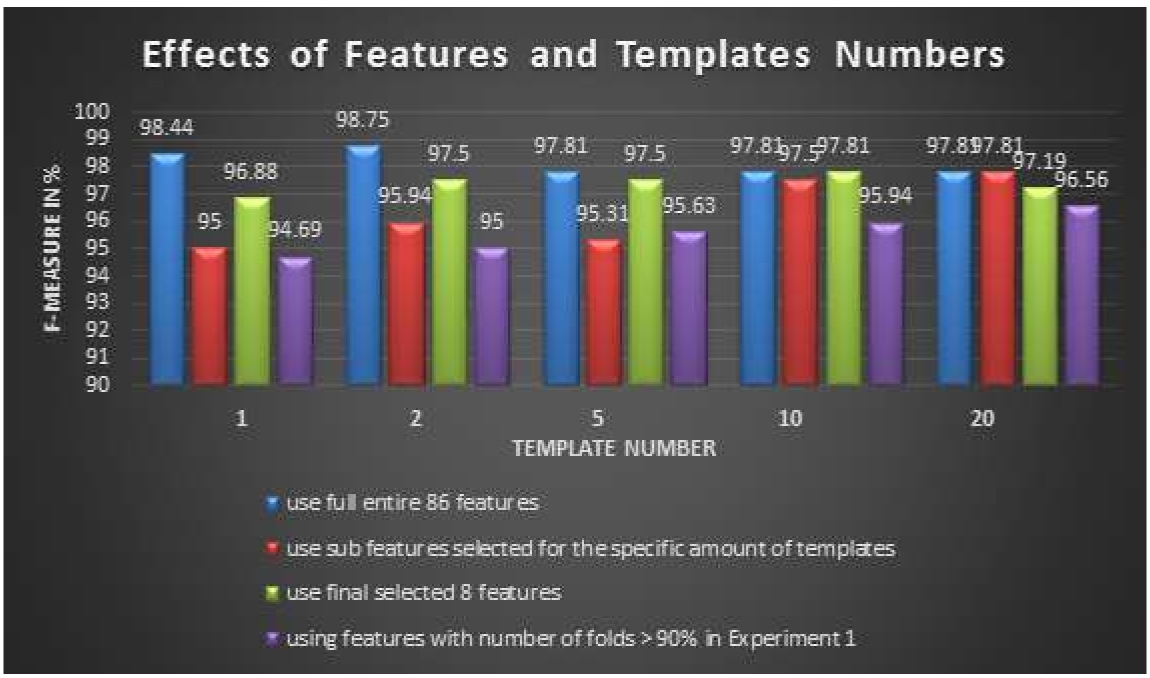


Figure 45 Compaison of f-measure for various template numbers and features

Therefore, we did feature selection under 1, 2, 5, and 10 templates as well to see how different features would be selected if the templates change. Not surprisingly, they shared most of the features, however, the main difference is that more template features are selected, for instance in 5 templates scenario, 5 out of 10 features selected are about hausdorff or OneDollar confidence feature. Reruning the recognition, the accuracy and f-measures are all decreased, especially for few templates, shown as red bar in Figure 45.

Experiment (2) is to run the feature selection with 10 fold cross validation. Table 15 represents the result of how many times each feature is selected. Note that most of the features are not selected in a single run. In this experiment, we select the features which have been chosen over 90% times. In totally, there are 14, which are in bold in Table 15. However, the result turns out not that satisfying, shown as purple bar in Figure 45, which performs close or even worse than that in last experiment.

Therefore we had our last feature selection experiment – to select features recursively until it becomes stable, i.e. no features are ignored any longer. Finally, we had 8 features in red in Table 15. Applying this features again, we are glad to see it achieves better performance than experiment (1) and (2) generally. Balancing the accuracy and template number trade off, the optimal solution is using the optimal subset and 2 templates as our final recognition scheme, which is circled in yellow in Figure 45.

Table 15 Feature selection results

1. f1 0 (0%)	**19. f19 10 (100%)**	37. rubine_Pie 0 (0%)	55. hausdorff_HengZhe Ti 0 (0%)	73. oneDollar_ShuZheZheGou 0 (0%)
2. f2 0 (0%)	**20. f20 10 (100%)**	38. rubine_Na 2 (20%)	56. hausdorff_ShuZhe 0 (0%)	74. oneDollar_Heng 0 (0%)
3. f3 0 (0%)	21. f21 2 (20%)	39. long_HengZheTi 0 (0%)	57. hausdorff_ShuZheZ heGou 0 (0%)	75. oneDollar_Ti 0 (0%)
4. f4 2 (20%)	22. f22 0 (0%)	**40. long_ShuZhe 10 (100%)**	58. hausdorff_Heng 0 (0%)	**76. oneDollar_HengGou 9 (90%)**
5. f5 0 (0%)	23. rubine_HengZheTi 0 (0%)	**41. long_ShuZheZheGou 9 (90%)**	59. hausdorff_Ti 0 (0%)	77. oneDollar_Gou 1 (10%)
6. f6 2 (20%)	24. rubine _ShuZhe 2 (20%)	42. long_Heng 2 (20%)	60. hausdorff_HengGou 0 (0%)	78. oneDollar_HengZhe 0 (0%)
7. f7 0 (0%)	25 rubine _ShuZheZheGou 0 (0%)	43. long_Ti 0 (0%)	61. hausdorff_Gou 0 (0%)	79. oneDollar_ShuWanGou 0 (0%)
8. f8 0 (0%)	26. rubine_Heng 4 (40%)	44. long_HengGou 0 (0%)	62. hausdorff_HengZhe 0 (0%)	80. oneDollar_HengZheWanGou 0 (0%)
9. f9 0 (0%)	27. rubine_Ti 0 (0%)	45. long_Gou 0 (0%)	63. hausdorff_ShuWan Gou 0 (0%)	81. oneDollar_Dian 0 (0%)
10. f10 0 (0%)	28. rubine_HengGou 0 (0%)	46. long_HengZhe 0 (0%)	64. hausdorff_HengZhe WanGou 1 (10%)	82. oneDollar_Shu 0 (0%)
11. f11 0 (0%)	29. rubine_Gou 2 (20%)	47. long_ShuWanGou 3 (30%)	65. hausdorff_Dian 0 (0%)	83. oneDollar_HengPie 20 (20%)
12. f12 2 (20%)	30. rubine_HengZhe 0 (0%)	48. long_HengZheWanGou 0 (0%)	66. hausdorff_Shu 0 (0%)	84. oneDollar_Wan 0 (0%)
13. f13 10(100%)	31. rubine_ShuWanGou 2 (20%)	49. long_Dian 0 (0%)	**67. hausdorff_HengPie 10 (100%)**	85. oneDollar_Pie 0 (0%)
14. f14 0 (0%)	32. rubine_HengZheWanGou 0 (0%)	50. long_Shu 0 (0%)	68. hausdorff_Wan 0 (0%)	86. oneDollar_Na 0 (0%)
15. f15 2 (20%)	33. rubine_Dian 0 (0%)	51. long_HengPie 0 (0%)	**69. hausdorff_Pie 9 (90%)**	
16. f16 2 (20%)	34. rubine_Shu 1 (10%)	**52. long_Wan 9 (90%)**	70. hausdorff_Na 0 (0%)	
17. f17 2 (20%)	**35. rubine_HengPie 10 (100%)**	**53. long_Pie 10 (100%)**	71. oneDollar_HengZhe Ti 0 (0%)	
18. f18 0 (0%)	36. rubine_Wan 0 (0%)	54. long_Na 0 (0%)	72. oneDollar_ShuZhe 2 (20%)	

6. FUTURE WORKS

We proposed a sketch based Chinese writing learning tool and help students understand the structure of Chinese characters while practicing. However, there are still several improvements to finish, and features to add on it until it can finally replace the traditional classes.

Firstly, it does not include interaction between students and teachers, such as grading, and reviewing, which could be a very helpful feature. Similarly, even if it automatically provides feedbacks, it is not automatically grading. To do this, we could add a teacher mode, which allows teachers to review and replay the students drawing; to provide grades and feedbacks manually; to edit problem more easily; and finally to generate questions and problem definition XML automatically.

Secondly, to improve the recognition, as discussed in this paper, we could invent and research on better features. Through feature selections, we already know several interesting results, such as template features are not as important as mathematical features, and so on. We believe more specific features such as "end-hookness", "concave-convexness", "corner number" will definitely become more efficient in recognizing Chinese stroke shapes. Thus the recognition will be reduced and simplified.

Finally, one other direction to improve the tool is to develop the web mode, which allows this tool transplantable from device to device, system to system. Windows 8, MacOS, Android, iOS, Linux, etc. There shall be no compatible issues. It also makes the tool accessible to much broader students and become a useful MOOC.

7. CONCLUSIONS AND CONTRIBUTIONS

In this paper, we investigated the Chinese character structure. Understanding relationship between the stroke, radical, and character is much more important and efficient than just memorizing stroke combinations.

Furthermore, we implemented a compatible recognition scheme to recognize Chinese characters hierarchically. Recognizing basic and compound Chinese strokes as primitives, the multilayer perceptron recognizer took advantages both of visual similarity information from template recognizer, and of writing technique information from feature recognizer. It improved stroke recognition accuracy by 15.7% than the average of the four basic recognizer.

Additionally, we analyzed feature effect and template number effect on the recognition results. Experiment results demonstrates that majority of the features are overlapping. The most important features include "the aspect of the bounding box", and the "density metrics", and "curviness". Finally, we chose 8 most important features after the recursive selecting stabilized. In most situations, feature recognition is more important than template recognition. The reason could be that writing technique is emphasized while they are taught. Therefore, 2 templates works as well as 20 templates, which is a good news since it improve recognition speed.

On the other hand, we implemented the concept of LADDER. We designed constraints of characters hierarchically for recognition, and feedbacks about writing techniques including stroke looking, stroke order, and component order. User study showed that the feedbacks are clear and instructive which helps memorize and understand.

Lastly, we developed a learning interface applying the recognition scheme we proposed, along with ladder to segment strokes and recognize higher level radicals and characters. In conclusion, Chinese Calligraphist is a sketch based Chinese language learning tool for western students, which integrates with paper-like writing environment, practicing session, writing guidance, recognition, and personal and timely feedbacks.

REFERENCES

[1] C. Alvarado, “Dynamically Constructed Bayes Nets for Multi-Domain Sketch Understanding”, *Proc. ACM SIGGRAPH 2006 Courses Article*, n. 32, 2006

[2] C. Alvarado, “SketchREAD: A Multi-Domain Sketch Recognition Engine”, *Proc. 17th Annual ACM Symposium on User Interface and Technology*, pp. 23-32, 2004

[3] M. Field and S. Valentine, “Sketch Recognition Algorithms for Comparing”, *Proc. IJCAI ‘11*, pp. 2436-2441, 2011

[4] M. Field and S. Valentine, “Sketch Recognition Algorithms for Comparing Complex and Unpredictable Shapes”, *Proc. 22nd International Joint Conference on Artificial Intelligence*, pp. 2436-2441, 2011

[5] T. Hammond, “A Sketch Recognition System for Recognizing Free-Hand Course of Action Diagrams”, *IAAI*, 2010

[6] T. Hammond, “LADDER: A Perceptually-based Language to Simplify Sketch Recognition User Interface Development”, PhD Dissertation, MIT, Cambridge, 2007

[7] T. Hammond and R. Davis, “LADDER, a Sketch Language for User Interface Developers” *Proc. Computers and Graphics*, v. 29, n. 4, pp. 518-532, 2005

[8] J. Johnson and T. Hammond, “Computing Confidence Values for Geometric Constraints for Use in Sketch Recognition." *Proc. of the Seventh Sketch-Based Interfaces and Modeling Symposium. Eurographics Association*, 2010

[9] L. B. Kara, “An Image-based, Trainable Symbol Recognizer for Handwritten Sketches”, *Proc. Computer and Graphics*, v29, pp. 510-517, 2004

[10] Y. Li, “Protractor: A Fast and Accurate Gesture Recognizer”, *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pp. 2169-2172, 2010

[11] Y. Li, H. Yang, J. Xu, W. He and J. Fan, “Chinese Character Recognition Method Based on Multi-Features and Parallel Neural Network Computation”, *Proc. Intelligent Computing 3rd International Conference on Advanced Intelligent Computing Theories and Applications*, pp. 1103-1111, 2007

[12] C.-L. Liu, “Online Recognition of Chinese Characters: The State-of-the-Art”, *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence,* v. 26, pp. 198-213, 2004

[13] AC. Long, "Visual Similarity of Pen Gestures", *Proc. SIGCHI Conference on Human Factors in Computing Systems '00*, pp. 360-367, 2000

[14] L. Ma, "A Study of Feature Design for Online Handwritten Chinese Character Recognition based on Continuous-Density Hidden Markov", *Proc. 10th International Conference on Document Analysis and Recognition,* pp. 49-54, 2009

[15] B. Paulson and T. Hammond, "PaleoSketch: Accurate Primitive Sketch Recognition and Beautification", *Proc. 13th International Conference on Intelligent User Interfaces*, 2008

[16] B. Paulson and T. Hammond, "What!? No Rubine Features: Using Geometric-based Features to Produce Normalized Confidence Values for Sketch Recognition", *Proc. VL/HCC Sketch Tools for Diagramming Workshop,* pp. 57-63, 2008

[17] R. Peterson, "Kanji Storyteller: Learn Kanji While Telling the World a Fantastic Graphic Story", https://code.google.com/p/kanji-storyteller, June, 2013

[18] D. Rubine, "Specifying Gestures by Example", *Proc. 18th Annual Conference on Computer Graphics and Interactive Techniques,* v. 25, i. 4, pp. 329-337, 1991

[19] D. Rubine, "The Automatic Recognition of Gestures", PhD Dissertation, Carnegie Mellon University, Pittsburgh, 1991

[20] T. M. Sezgin, "Sketch based Interfaces: Early Processing for Sketch Understanding", *Proc. Workshop on Perceptive User Interface*, 2001

[21] P. Taele, "Freehand Sketch Recognition for Computer-Assisted Language Learning of Written East Asian Languages", Master Thesis, Texas A&M University, College Station, 2010

[22] P. Taele and T. Hammond, "Hashigo: A Next-Generation Sketch Interactive System for Japanese Kanji", *Proc. 21st Innovative Application of Artificial Intelligence Conference*, pp. 153-158, 2009

[23] J. O. Wobbrock, "Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes", *Proc. 20th Annual ACM Symposium on User Interface Software and Technology*, pp. 159-168, 2007

[24] A. Wolin, B. Eoff, and T. Hammond, "ShortStraw: A Simple and Effective Corner Finder for Polylines", *Proc. 5th Eurographics Conference on Sketch-Based Interfaces and Modeling*, pp. 33-40, 2008

[25] B. Yu, "A Domain-Independent System for Sketch Recognition", *Proc. 1st International Conference on Computer Graphics and Interactive Techniques in*

Australasia and South East Asia, pp. 141-146, 2003

[26] M. F. Zafar, D. Mohamad, and R.M. Othman, "On-line Handwritten Character Recognition: An Implementation of Counterpropagation Neural Net," *World Academy of Science Engineering and Technology*, vol. 10, pp. 232-237, 2005

APPENDIX A

LESSON MATERIALS IN USER STUDY

Lesson 0:
Howdy, welcome to lesson 0. In this class, you will learn some basic Chinese writing concept. Basic strokes, several compound strokes, radicals, as well as several simple Chinese characters. Let's start!

Basic Strokes	Stroke Name	Stroke Pronunciation	Example
一	横	**Heng (H)**	二 **(two)**
丨	竖	**Shu (S)**	十 **(ten)**
丿	撇	**Pie (P)**	大 **(big)**
㇏	捺	**Na (N)**	人 **(people)**
丶	点	**Dian (D)**	广 **(broad)**
㇀	提	**Ti (T)**	江**(river)**

Compound	Stroke Name	Stroke Pronunciation	Example
㇖	横钩	**HengGou (HG)**	冗 **(redundant)**
㇇	横撇	**HengPie (HP)**	子 **(son)**
㇕	横折	**HengZhe (HZ)**	口 **(mouse)**
乙	横折弯钩	**HengZheWanGou (HZWG)**	九 **(nine)**
㇄	竖折	**ShuZhe (SZ)**	山 **(mountain)**
㇟	竖弯钩	**ShuWanGou (SWG)**	儿 **(child)**

Radicals	Name	Pronunciation	Stroke Number	Stroke Order
木	**Tree**	**Mu**	**4**	**H, S, P, N**
氵	**Water**	**SanDianShui**	**3**	**D, D, T**

Characters	Name	Pronunciation	Stroke Number	Radicals
林	**Grove**	**Lin**	**8**	**Tree, Tree**
森	**Forest**	**Sen**	**12**	**Tree, Tree, Tree**
又	**Again**	**You**	**2**	**Stroke （HP, N）**
汉	**Chinese**	**Han**	**5**	**Water, Again**

Problem Set: Please Write:
Stroke Heng: ___________ Stroke Pie: ___________ Stroke ShuZhe: ___________
Radical Water: __________ Radical Tree: __________ Character Chinese: ________

Lesson 1:
Howdy, with lesson 0, I hope you are a little bit familiar with Chinese writing. It is fun, isn't it? Okay, let's begin our real class. Lesson 1 teaches you how to writing Chinese numbers. Let's count!

Characters

Shape	Name	Pronunciation	Stroke Number	Radicals/Stroke Order
一	**One**	**Yi**	**1**	**H**
二	**Two**	**Er**	**2**	**H, H**
三	**Three**	**San**	**3**	**H, H, H**
四	**Four**	**Si**	**5**	**S. HZ, P, SZ, H**
五	**Five**	**Wu**	**4**	**H, S, HZ, H**
六	**Six**	**Liu**	**4**	**D, H, P, D**
七	**Seven**	**Qi**	**2**	**H, SWG**
八	**Eight**	**Ba**	**2**	**P, N**
九	**Nine**	**Jiu**	**2**	**P, HZWG**
十	**Ten**	**Shi**	**2**	**H, S**
Extras				
百	**Hundred**	**Bai**	**6**	**H, P, S, HZ, H, H**
千	**Thousand**	**Qian**	**3**	**P, H, S**
万	**Ten Thousand**	**Wan**	**3**	**H, P, HZ**
亿	**Hundred Million**	**Yi**	**3**	**P, S, HZWG**

Problem Set: Please Write:

2: __________ 3: _________ 5: _________ 7: ____________
11: _________ 13: _________ 17: _________ 19: ____________
23: _________ 29: _________ 31: _________ 37: ____________
1: __________ 4: _________ 9: _________ 16: ____________
25:_________ 36: __________ 49: ________ 64:____________
81: _________ 100:__________

Extras:

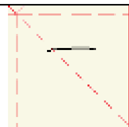
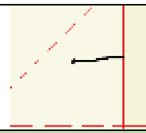
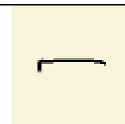
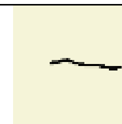
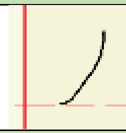
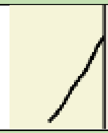
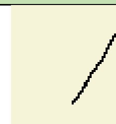
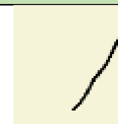
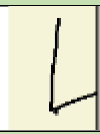
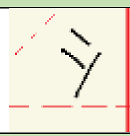
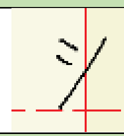
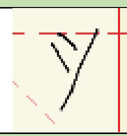
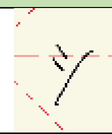
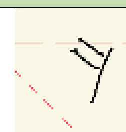
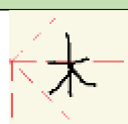
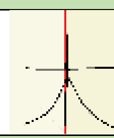
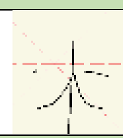
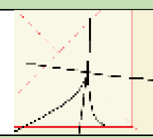
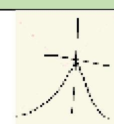
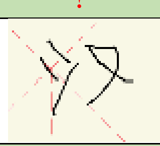
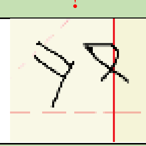
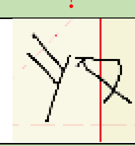
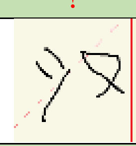
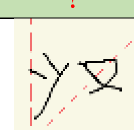
1: __________ 256: __________
2: __________ 512: __________
4: __________
8: __________
16: _________
32: _________
64: _________
128: ________

硬笔书法空白字帖

APPENDIX B

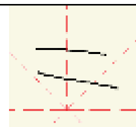
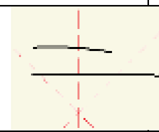
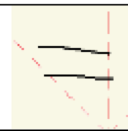
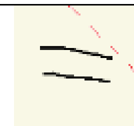
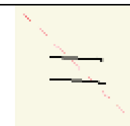
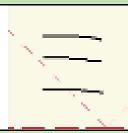
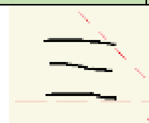
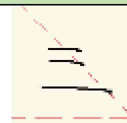
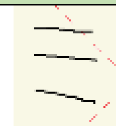
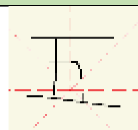
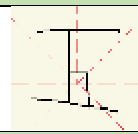
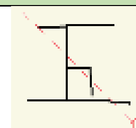
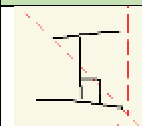
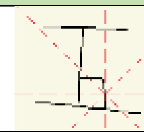
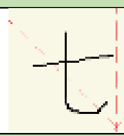
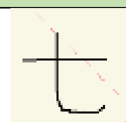
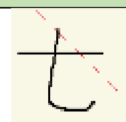
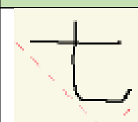
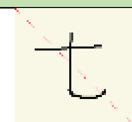
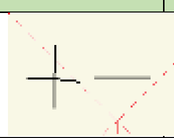
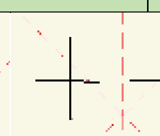
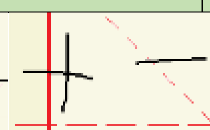
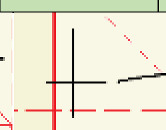
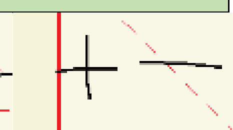
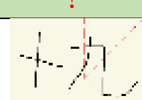
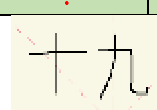
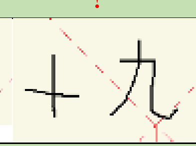
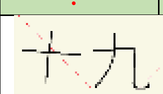
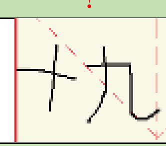
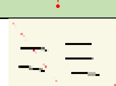
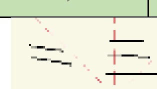
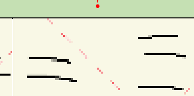
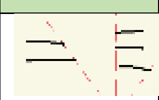
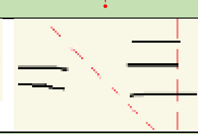
USER STUDY RESULTS

Student 1: Lesson 0

	1	2	3	4	5
Heng					
Visually	✓	✓	✓	✓	
Technically	✓	✓	!	!	
Pie					
Visually	✓	✓	✓	✓	
Technically	✓	✓	✓	✓	
ShuZhe					
Visually	×				
Technically	!				
Radical Water					
Visually	×	×	×	×	×
Technically	✓	✓	✓	✓	✓
Radical Tree					
Visually	✓	✓	✓	✓	✓
Technically	!	!	!	!	!
Character Chinese					
Visually	×	×	×	×	×
Technically	✓	✓	✓	✓	✓

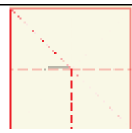
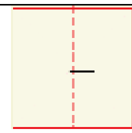
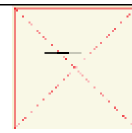
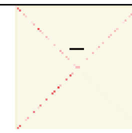
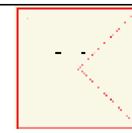
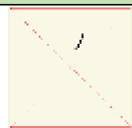
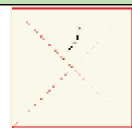
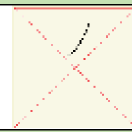
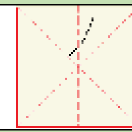
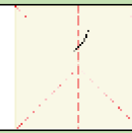
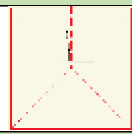
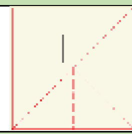
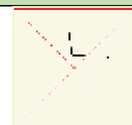
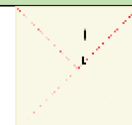
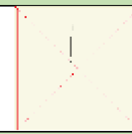
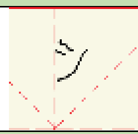
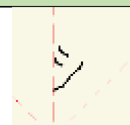
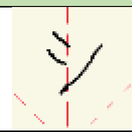
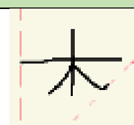
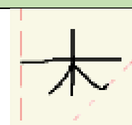
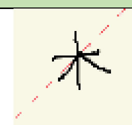
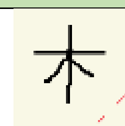
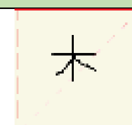
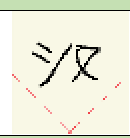
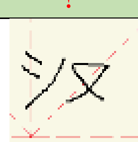
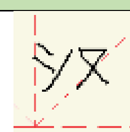
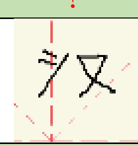
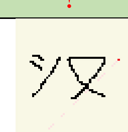
Technical feedbacks:
Heng: No. 4 is writing in negative slope
ShuZhe: No. 1 is written in two strokes
Radical Tree: Shu should be written after Heng. All of the 5 are writing Shu first.
Character Chinese: Last stroke of No. 2 is recognized as Dian

Student 1: Lesson 1

	1	2	3	4	5
Two					
Visually	✓	✓	×	×	✓
Technically	✓	✓	✓	✓	✓
Three					
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Five					
Visually	✓	×	×	×	×
Technically	✓	!	!	!	!
Seven					
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Eleven					
Visually	✓	✓	✓	✓	✓
Technically	!	!	!	!	!
Nineteen					
Visually	✓	✓	✓	✓	✓
Technically	!	✓	!	✓	!
Twenty Three					
Visually	×	×	✓	✓	×
Technically	✓	✓	✓	✓	✓

Technical feedbacks:
5: The third stroke HengZhe should exceed the Shu to the left.
11: the first character Ten, was in wrong stroke order. Heng should be written first
19: No. 1, 3, 5, the same.

Student 2: Lesson 0

	1	2	3	4	5
Heng					
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Pie					
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
ShuZhe					
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Radical Water					
Visually	×	×	×	×	×
Technically	✓	✓	✓	✓	✓
Radical Tree					
Visually	✓	✓	✓	×	✓
Technically	✓	!	✓	!	!
Character Chinese					
Visually	×	×	×	×	×
Technically	✓	✓	✓	✓	✓

Technically feedbacks:

Radical Tree: No. 2, 4, and 5, Shu was written before Heng, but Heng should be written at first

Student 2: Lesson 1

	1	2	3	4	5
Two	二	二	二	二	二
Visually	×	×	×	×	×
Technically	✓	✓	✓	✓	✓
Three	三	三	三	三	三
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Five	五	五	五	五	五
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Seven	七	七	七	七	七
Visually	✓	✓	✓	✓	✓
Technically	✓	✓	✓	✓	✓
Eleven	十一	十一	十一	十一	十一
Visually	✓	✓	✓	✓	✓
Technically	!	!	!	!	✓
Nineteen	十九	十九	十九	十九	十九
Visually	✓	✓	✓	✓	✓
Technically	!	✓	✓	✓	✓
Twenty Three	二十三	二十三	二十三	二十三	二十三
Visually	×	×	✓	✓	✓
Technically	!	!	!	!	!

Technically feedbacks:

11: No. 1 – 4, the first character Ten, was in wrong stroke order. Heng should be written first

19: No. 1, the same.

23: the same

Student 3: Lesson 1