A Chinese Handwriting Education System with Automatic Error Detection

Zhi-Hui Hu[1,2,3], Yun Xu[1,2], Liu-Sheng Huang[1,2] and Howard Leung[2,3]

[1]Department of Computer Science and Technology, University of Science & Technology of China, Hefei, China
[2]Joint Research Lab of Excellence, CityU-USTC Advanced Research Institute, Suzhou, China
[3]Department of Computer Science, City University of Hong Kong, Hong Kong, China
Email: kittyhu@mail.ustc.edu.cn; {xuyun, lshuang}@ustc.edu.cn; howard@cityu.edu.hk

Abstract—**We build a distance education application of a Chinese handwriting education system that allows students to do practice at anytime and anywhere. As an intelligent tutor, the system can automatically check the handwriting errors, such as the stroke production errors, stroke sequence error and stroke relationship error. Then our system should provide useful feedback to the student. In this paper, attributed relational graph matching is used to locate the handwriting errors. The pruning strategy is applied to reduce the computational time. The experiment results show that our proposal can handle more handwriting error cases than existing methods with a higher accuracy.**

Index Terms—**Chinese handwriting education, handwriting errors, automatic error detection, intelligent tutoring, attributed relational graph matching.**

I. Introduction

In a traditional Chinese handwriting class, the teacher should first writes a Chinese character on the blackboard, and then students rewrite the character on their exercise book by following the teacher's handwriting. After the class, teacher has to check for handwriting errors in students' handwriting. Finally, the students can get the feedback of their exercise in the next class. There are mainly three drawbacks of the traditional handwriting education. 1) Time-consumption. Students cannot get an immediate feedback about their handwriting exercise in class, because teacher needs a lot of time to check for the handwriting errors; 2) Faultiness. From the exercise book, teacher cannot check the correctness of the stroke order and certain kinds of handwriting errors. However, Law et al. [1] found that students often make many handwriting mistakes including stroke sequence error and stroke production error. Meanwhile, stroke relationship error is reported in [2]; 3) Teacher-oriented. Without a teacher, students can hardly learn the handwriting by themselves. Moreover, the workload for the teacher is high. This motivates us to build a Chinese handwriting education system to assist teacher in automatically checking the handwriting error, then students can get the feedback immediately whenever and wherever he/she wants to do practice.

Similar to handwriting recognition, the problem in handwriting education involves extracting the features of the character at first. After feature extraction, the handwriting education requires different analysis methods from handwriting recognition. Handwriting recognition is focused on classifying the input character into one of the candidate character with the highest similarity [3][4][5]. On the other hand, handwriting education requires a detailed matching between the input and template characters in order to locate the exact differences. It is a difficult problem to locate the exact handwriting errors due to the complex structure of Chinese characters, widely variable writing styles as well as numerous combinations of possible errors.

Some related works have been done on e-learning system [6][7] which aims to assist students to get more useful advice in their distance education. They had developed an intelligence tutoring education system to provide a ubiquitous learning environment to the students. With the development of pen-based devices, it is now possible to apply e-learning techniques to handwriting education. Several Chinese handwriting education systems have been provided to assist students to learn the writing of Chinese characters. Some researchers focus on the view-only handwriting system so students cannot do practice through the system [8][9]. Other researchers allow students to do practice and then the system can detect the handwriting errors. Some of these approaches only work on checking the stroke sequence error [10]. Others can locate the stroke production error [11][12][13]. Meanwhile, the method in [14] can find out the stroke relationship error. The work proposed in [17] can find both the stroke production error and stroke sequence error but without considering the spatial relationship errors.

In our handwriting education system, template characters can be shown in a stroke by stroke mode. Then the students can follow it to rewrite the character. After that, with the automatic error detection tool the system can check the handwriting errors in students' handwriting and provide useful feedback to student.

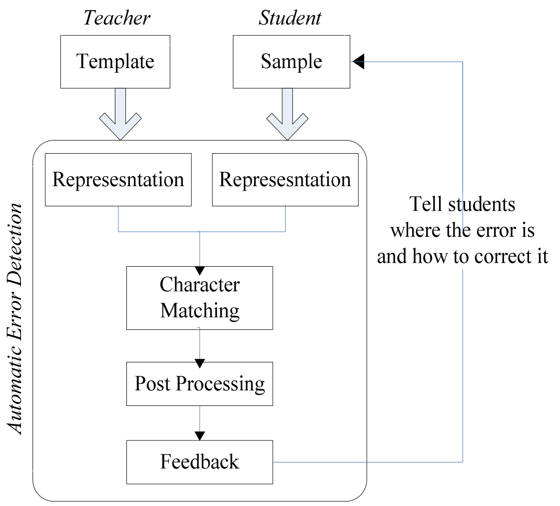


Figure 1. Our Chinese handwriting education system

In this paper, the attributed relational graph (ARG) matching [18][19][20] is applied to locate the stroke production errors and stroke sequence error. Then post processing is used to find out the spatial relationship errors between strokes. Finally, the students can get some useful feedback about where the handwriting errors are and how to correct them.

The remainder of the paper is organized as follows: In Section Ⅱ, our proposed system is described. Then, section Ⅲ shows our automatic error detection tool. The experiments and results are discussed in Section Ⅳ. Finally, Section Ⅴ is devoted to the conclusions and future work.

II. Our Proposed System

The flow of our handwriting education system is illustrated in Fig. 1. First, the sample handwriting which the student has inputted and the template character which the student has to follow are both represent as ARGs. Second, the error-tolerant graph matching is applied on the two ARGs to detect the stroke sequence error and stroke production error. Then post processing is used to locate the stroke relationship error. As a result, feedback is provided to students to let them know where the handwriting error is and how to correct it.

We have implemented our handwriting education system with the World Wide Web technology. The system is available online through the webpage http://vache.cs.cityu.edu.hk/ccls/. It is designed for both teachers and students to use and it contains an automatic error detection module.

A. Teacher

Once the teacher writes a Chinese character the system will automatically store it as the template character (Fig. 2). Those template characters are used to be followed by student while they do their handwriting exercise. Then, the teacher can choose any of the characters in the database to create a handwriting exercise for student.

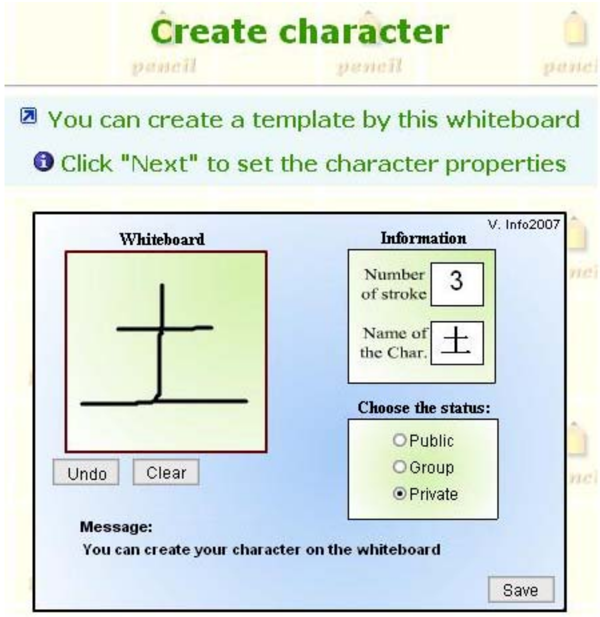


Figure 2. Teacher creates template character

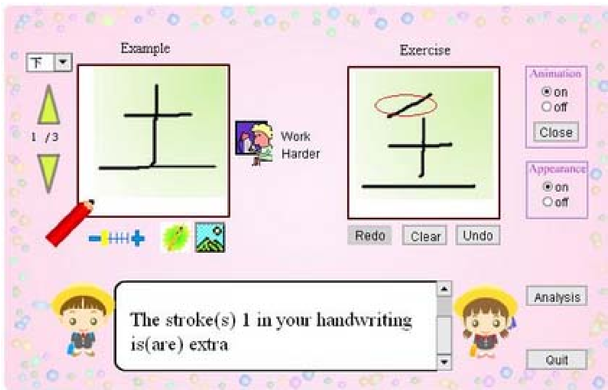


Figure 3. Handwriting error detection

B. Student

The student can choose any of the handwriting exercise created by the teacher for practice. The template character will be displayed with the stroke by stroke animation. Then the student does the handwriting practice by following it. After that, the student can get an immediate feedback to show whether there are some handwriting errors. As a result, student can practice handwriting whenever and wherever he/she wants.

C. Automatic error detection

After student submits their sample character, the automatic error detection can immediately locate those errors in the sample character with the red mark and provide feedback to further explain the handwriting error (Fig.3). The details of the automatic error detection will be described in the following section.

III. Automatic Error Detection

In our Chinese handwriting education system, the automatic error detection aims to locate the handwriting errors in student's handwriting and provide immediate feedback.

Relation	Symbol	Symbol for inverse	Example
a before b	<f	>l	
	<m	>m	
	<l	>f	
a meets b	m	mi	
a overlaps b	of	oil	
	om	oim	
	ol	oif	
a equals b	=	=	
a starts b	sf	sif	
	sm	sim	
	sl	sil	
a during b	df	dif	
	dm	dim	
	dl	dil	
a finishes b	ff	fif	
	fm	fim	
	fl	fil	

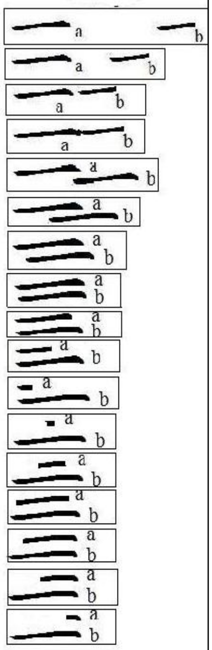


Figure 4. Refined interval relationship

A. Representation

ARG was first described in [18] to represent the structural information of a pattern. Let V be the set of nodes and E be the set of the edges. Then the graph can be represented as $g = (V,E,\alpha,\beta)$, where $\alpha : V \rightarrow L_V$ is the node labeling function, $\beta : E \rightarrow L_E$ is the edge labeling function, L_V and L_E are the sets of node and edge labels respectively.

In our application, the set of nodes V is used to describe the strokes of the Chinese character and the set of edges E describes the relationships [14] between any two strokes as defined by our refined interval relationship in Fig.4. The complete ARG representation for a Chinese character is given as follows.

Nodes in the ARG: Each node stores the x and y coordinates of a stroke, i.e., the attributes of the node a is $a=(x_i, y_i)$, where $i=1,2,\dots,n$. The node labeling function $\alpha : V \rightarrow L_V$ returns the n data points for each stroke.

Edges in the ARG: Each edge stores the relationship of the two nodes (strokes) which are connected by this edge. The edge labeling function $\beta : E \rightarrow L_E$ returns (μ, λ) where μ is the refined interval relationship (Fig.4) along the x-axis and λ is the refined interval relationship along the y-axis[16].

As an example, a Chinese character and the spatial relationships between strokes are shown in Fig.5(a). The ARG representation of this Chinese character is shown in Fig.5(b). The strokes *a, b* and *c* in the character are represented by the nodes *a, b* and *c* in the ARG. The variable r_{s1s2} is the relationship between strokes $s1$ and $s2$, and $s1, s2 \in (a,b,c), s1 \neq s2$.

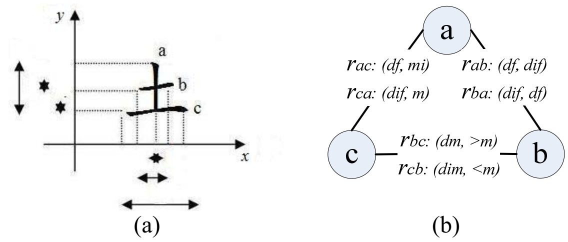


Figure 5. ARG representation of a Chinese character. (a) A Chinese character. (b) Corresponding ARG

The relationships r_{ac} is denoted by (df ,mi), r_{ab} is denoted by (df, dif), and r_{bc} is denoted by (dm,>m). Note that the r_{ca} is formed simply by taking the inverse of each component of the relationship used to represent r_{ac} and is denoted by (dif, m).

B. Character Matching

As illustrated in Fig. 1, each of the template character and sample character is represented as an ARG. The input (sample) ARG $g_1 = (V_1, E_1, \alpha_1, \beta_1)$ represents the input character and the template ARG $g_2 = (V_2, E_2, \alpha_2, \beta_2)$ represents the template character. In order to decide whether the two ARGs have some differences, we find an error-tolerant graph matching from g_1 to g_2 which is a function $f : \hat{V}_1 \rightarrow \hat{V}_2, \hat{E}_1 \rightarrow \hat{E}_2$, where $\hat{V}_1 \subseteq V_1$, $\hat{V}_2 \subseteq V_2$, $\hat{E}_1 \subseteq E_1$ and $\hat{E}_2 \subseteq E_2$. For our application, we apply a transformation, denoted by function f , from the input graph g_1 to the template graph g_2 . This function f consists of many edit operations performed on both nodes and edges.

The node operations have been defined by the authors in [9]: 1) *node substitution* implying that the input stroke is correct; 2) *node merging* implying that the input strokes are broken strokes; 3) *node splitting* implying that the input stroke is a concatenated stroke; 4) *node deletion* implying that the input stroke is an extra one; 5) *node insertion* implying that there is a missing stroke.

On the other hand, we get the definition of the edge operations as follows: 6) *edge substitution* implying that both nodes sharing this edge are correct; 7) *edge deletion* implying that one of the nodes sharing this edge is an extra or broken stroke, or both nodes sharing the same edge are extra or broken stroke; 8) *edge insertion* implying that one of the node who sharing this edge is a missing or concatenated stroke, or both nodes sharing the same edge are missing or concatenated stroke. It can be seen that these edit operations can identify the stroke production errors in the input handwriting.

Edge substitution: The cost for the edge substitution is the matching cost between an edge in the sample character and an edge in the template character. We use Rt to denote the set of edges in the template and Rs to the sample. The i-th template edge Rt_i can be denoted by $(\mu t_i, \lambda t_i)$ and the j-th sample edge Rs_j can be denoted by $(\mu s_j, \lambda s_j)$. The dissimilarity between $(\mu t_i, \lambda t_i)$ and $(\mu s_j, \lambda s_j)$ is defined as $D(Rt_i, Rs_j)$ which is derived from the idea of the interval neighborhood graph [15][21].

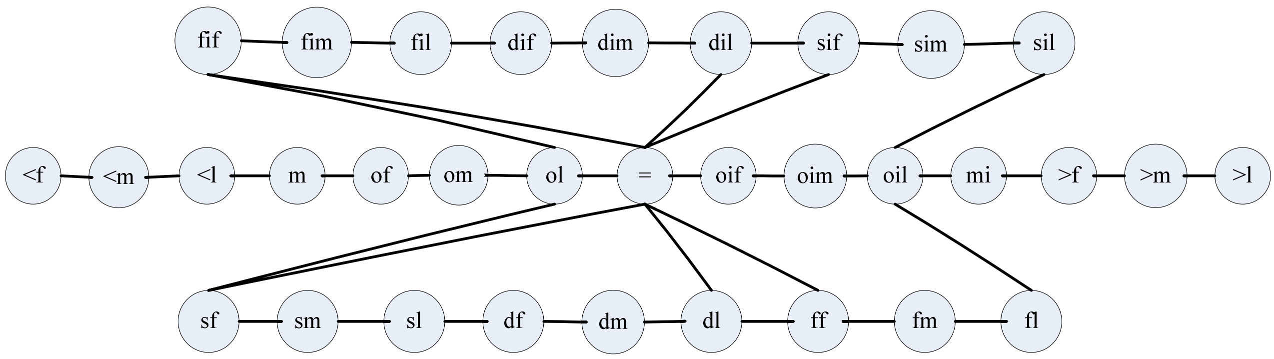


Figure 6. Refined interval neighborhood graph

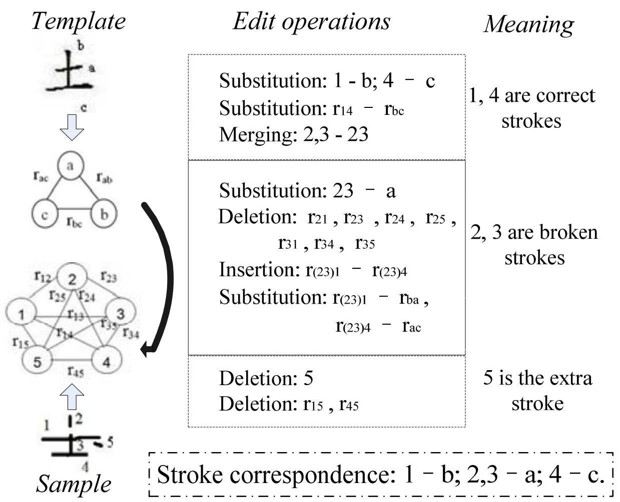


Figure 7. Examples for ARG matching with edit operations

Two interval relationships are neighbors, if they can be transformed into one another by continuous deformation (shortening, lengthening, and moving)[21]. We construct a new interval neighborhood graph in Fig.6 which considers our proposed refined relationship with three levels (*f, m, l*) in each relationship defined in Fig.4. Note that the three levels with the same interval relationship are close to each other in the refined interval neighborhood graph since they can be transformed from one to another by shortening or lengthening the distance between the two strokes.

The distance between two interval relationships μt_i, and μs_j or the distance between λt_i and λs_j is defined as the topological distance between the two relationships, i.e., the length of the shortest path from μt_i to μs_j or from λt_i or λs_j in the interval neighborhood graph. The final spatial relationship distance $D(Rt_i, Rs_j)$ can be defined as follows:

$$D(Rt_i, Rs_j) = \sqrt{D(\mu t_i, \mu s_j)^2 + D(\lambda t_i, \lambda s_j)^2} \quad (1)$$

Edge deletion: It is used together with the node deletion operation. If some extra nodes need to be removed by using the node deletion, then the corresponding edge should also be removed by the edge deletion.

Edge insertion: It is similar to the edge deletion, but it is used together with the node insertion operation. If some missing nodes need to be added by using the node insertion, then the corresponding edge should also be added by the edge insertion.

The graph edit distance defines the overall cost for transforming from the ARG g_1 to ARG g_2 using the function *f* is defined as follows:

$$Cost(f, g_1, g_2) = C_{node}(f, g_1, g_2) + C_{edge}(f, g_1, g_2) \quad (2)$$

where C_{node} is the node edit distance and C_{edge} is the edge edit distance.

$$C_{node}(f, g_1, g_2) = C^n_{sub} + C^n_{mer} + C^n_{spl} + C^n_{del} + C^n_{ins} \quad (3)$$

where C^n_{sub}, C^n_{mer}, C^n_{spl}, C^n_{del}, C^n_{ins} are the costs of node substitution, merging, splitting, deletion and insertion respectively.

$$C_{edge}(f, g_1, g_2) = C^e_{sub} + C^e_{del} + C^e_{ins} \quad (4)$$

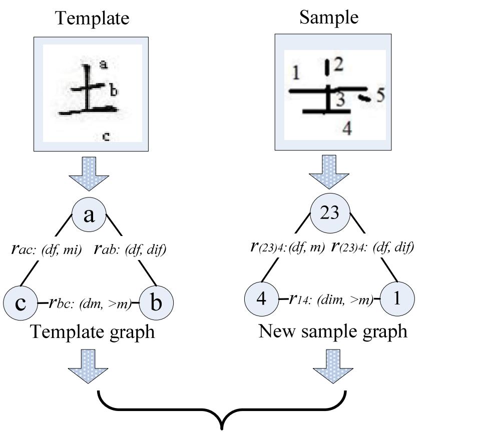
where C^e_{sub}, C^e_{del}, C^e_{ins} are the costs of edge substitution, deletion and insertion respectively.

An example of the optimal matching between two ARGs with the edit operations is described in Fig.7. Note that each edit operation has the corresponding meaning such that one can identify the exact handwriting errors by examining the edit operations.

There are many possible sequences of edit operations that can transform from g_1 to g_2. It is impractical to perform exhaustive search in order to find the optimal matching that minimizes the overall graph edit distance. As a result, the A* algorithm which is a state-space search strategy is employed in order to identify the optimal ARG matching. At the search level *n* the evaluation function *h(n)* is:

$$h(n) = h_1(n) + h_2(n) \quad (5)$$

where $h_1(n)$ is the actual matching cost from the initial search level to the current search level *n* , and $h_2(n)$ is the estimation of the matching cost from the search level *n* to the goal. For the A* algorithm, the inequality $h_2(n) \leqslant h_2^*(n)$ should hold for any search level *n*, where $h_2^*(n)$ is the actual matching cost from the current search level to the goal.



	r_{ac}-$r_{(23)4}$	r_{ab}-$r_{(23)1}$	r_{bc}-r_{14}
$D(\mu t_i, \mu s_i)$	0	0	4
$D(\lambda t_i, \lambda s_i)$	0	0	0

Figure 8. Post processing on finding the relationship error

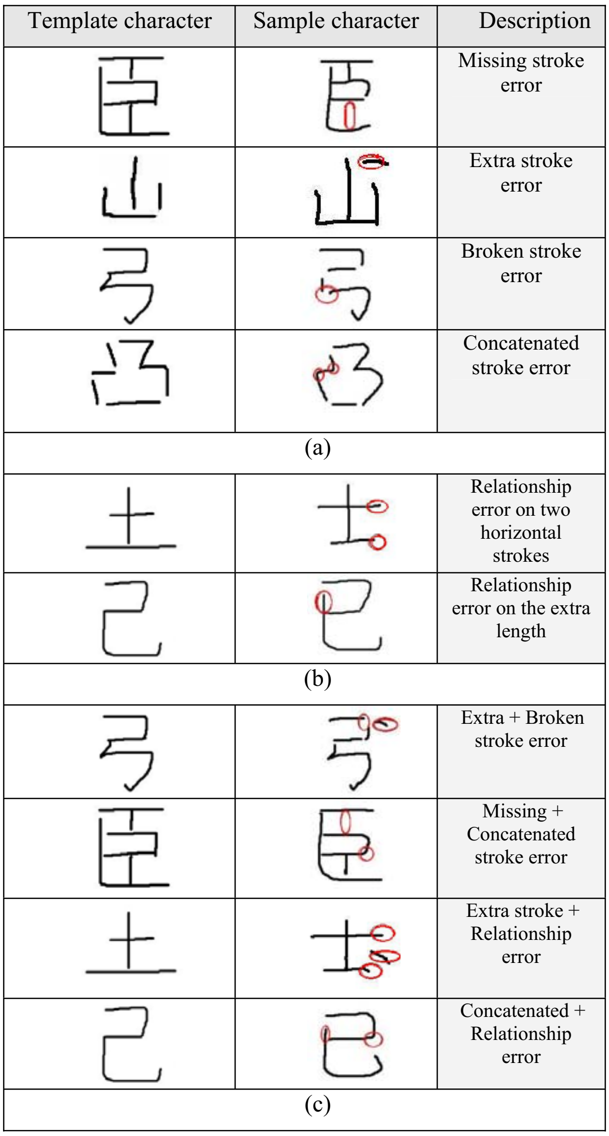


Figure 9. Feedback on handwriting error. (a) Stroke production error. (b) Stroke relationship error. (c) Multiple handwriting error.

C. Post Processing

After the ARG matching we can get a mapping from the sample ARG to the template ARG showing the stroke correspondence between the sample character and the template character. For the example in Fig. 7, after applying the resulting edit operations, the stroke correspondence between the sample and the template is: 1-*a*; 2,3-*b*; 4-*c* (5 is extra stroke in sample). We can then obtain a new graph to represent the sample graph as shown in Fig. 7. The edges in the graph can describe the spatial relations between the nodes (strokes). From the transformations, we can identify the distance between the corresponding spatial relations as illustrated in Fig. 8.

In Fig.8, there is a difference between the spatial relationships r_{bc} and r_{14}. In particular, the spatial relationship along the *x*-axis for r_{bc} is *dm* and the spatial relationship along the *x*-axis for r_{14} is *dim*. It can be found from Fig.6 that it takes at least 4 steps to get from the *dm* node to the *dim* node thus the spatial relationship distance between r_{bc} and r_{14} is 4. Since this is quite a large distance, we can conclude that the spatial relationship between strokes 1 and 4 is incorrect compared with the corresponding template strokes *b* and *c*. Then, there must be some kinds of relationship error between those two strokes.

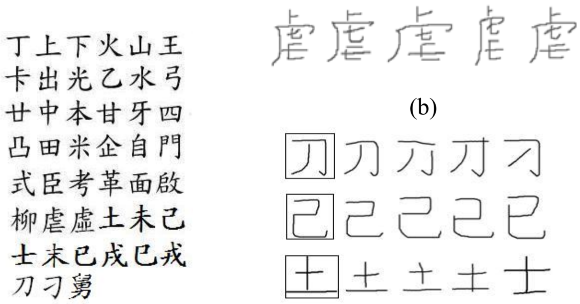
D. Feedback

After the character matching we can find out the stroke correspondence between the template and sample characters. If the student has made any stroke sequence error, we can verify the stroke order by the stroke correspondence. In addition to stroke sequence error, stroke production error in Fig.9(a) can also be located by character matching. Then post processing is used to detect the relationship errors in Fig.9(b). Our handwriting education system can detect multiple stroke handwriting errors as shown in Fig.9(c). Finally, useful feedback is revealed by circling the locations of the problematic strokes in red color, and the suggestion on how to correct the handwriting error is also provided.

IV. EXPERIMENTAL RESULTS

In our experiment, 44 Chinese characters shown in Fig.10(a) have been used. In this dataset, there are some simple characters which are used most frequently, and also, some character with more complex structure. We asked different people to write them, and then, we get totally 1247 various Chinese handwriting characters.

Different people may write the same character in different ways. The variation of the characters is illustrated in Fig.10(b) and Fig.10(c). In Fig.10(b), those variations are acceptable as there are no handwriting errors in it. However, in Fig.10(c), the variation from the character in the first column to the last one is very large. Although they look similar on their appearance they are completely different characters. Our system aims to identify those variations and notify the student about those problems.



(a) (b) (c)

Figure 10. The dataset. (a) Characters used in the dataset. (b) Acceptable variations of the characters. (c) Unacceptable variations of the characters.

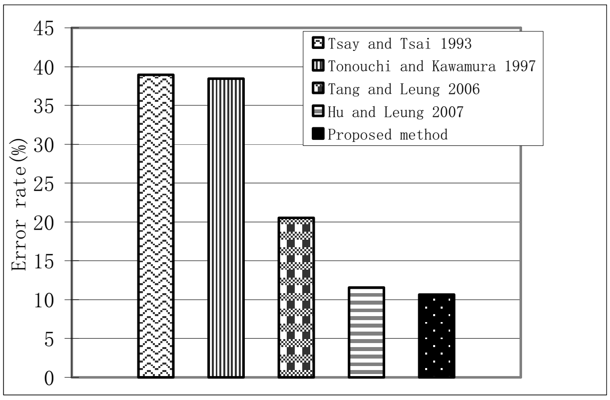


Figure 11. Performance of finding stroke production errors

Some of the people's handwritings contain errors and we manually check those error types to obtain the ground truth information. Our algorithm is then applied to identify handwriting errors. Afterwards we can compare the result returned by our proposed method with the ground truth information to evaluate the accuracy of our method.

We have compared our proposed method with four existing methods: Tsay and Tsai [5], Tonouchi and Kawamura [6], Tang and Leung [10] and Hu and Leung [7]. The methods in [5] and [6] are based on string matching which is sequence dependent. If the student input character has a different stroke sequence from the template, these methods may fail to find the matching. Tang and Leung [10] proposed a system that allows students to practice handwriting freely, and it can check both the stroke sequence error and stroke production errors simultaneously. However it relies heavily on some threshold values to determine the potential production errors. Hu and Leung [13] applied graph matching to find the stroke production errors, but without considering the sequence error and spatial relationship errors. On the other hand, our current proposed method is able to identify all three kinds of handwriting errors. The performance comparison can be seen in Fig.11.

V. CONCLUSION AND FUTURE WORK

In this paper we have proposed a Chinese handwriting education system with automatic error detection. We have used the attributed relational graph to represent a Chinese handwriting character incorporating the spatial relationship information between strokes. A refined interval relationship with more granular levels is proposed to model the Chinese characters. A novel interval neighborhood graph is also proposed to compute the distances among the refined interval relationships. The A* algorithm is used to find the optimal graph matching in order to locate the exact handwriting errors. Our system supports distance learning by allowing students to practice their handwriting through the web. The system can automatically check whether there are any stroke sequence, production and relationship errors in the student's handwriting, while existing methods cannot handle so many handwriting errors at the same time. With the automatic error detection, our system can assist the students to gain a more effective learning experience. As future work, we will continue to improve the accuracy in error detection. We also intend to check the handwriting error on more complex characters. In particular, we will work on the relationship error detection based on the characters with high similarities.

ACKNOWLEDGMENT

The work described in this paper was supported by a grant from the key project of the national nature science foundation of china a grant (Project No. 60533020), and a grant from City University of Hong Kong (Project No. 7001964).

REFERENCES

[1] N. Law, W.W. Ki, A.L.S. Chung, P.Y. Ko and H.C. Lam, "Children's stroke sequence errors in writing Chinese characters", *Reading and Writing: An Interdisciplinary Journal, Kluwer Academic Publishers*, pp. 267-292, 1998.

[2] C. K. Tan, "An algorithm for online strokes verification of Chinese characters using discrete features", *8th Intl. Workshop on Frontiers in Handwriting Recognition* (*IWFHR*), pp. 339–344, 2002.

[3] C. C. Tappert, C. Y. Suen and T. Wakahara, "The State of the Art in Online Handwriting Recognition", *IEEE Trans. PAMI*, vol. 12, no. 8, pp. 787-808, 1990.

[4] S. J. Lee and H. L. Tsai, "Pattern Fusion in Feature Recognition Neural Networks for Handwritten Character Recognition", *IEEE Trans, SMC*, vol.28, no.4, pp.612-617, August 1998.

[5] J. Q. Song, Z. Li, R. Lyu and S. J. Cai, "Recognition of Merged Characters Based on Forepart Prediction, Necessity-Sufficiency Matching, and Character-Adaptive Masking", *IEEE Trans, SMC*, vol.35, no.1, pp.2-11, February 2005.

[6] O. K. Zein and Y. Kermarrec, "Description and composition of e-learning", *Journal of software*, vol.2. no.5, pp.74-83, November 2007.

[7] K.Hiekata, "A framework for design engineering education with workflow-based e-learning system", *Journal of software*, vol.2. no.4, pp.88-95, October 2007.
[8] H. C. Lam, K. H. Pun, S. T .Leung, S. K. Tse and W. W. Ki. "Computer-Assisted-Learning for Learning Chinese Characters", *Communications of COLIPS*, *an Intl. Journal of the Chinese and Oriental Languages Processing Society*, vol. 3, no. 1, pp. 31–44, 1993.
[9] H. C. Lam, W. W. Ki, N. Law, A. L. S. Chung, P. Y. Ko, A. H. S. Ho and S. W. Pun. "Designing CALL for learning Chinese characters", *Journal of Computer Assisted Learning*, vol. 17, no. 1, pp. 115-128, 2001.
[10] K. T. Tang and H. Leung. "Reconstructing the correct writing sequence from a set of Chinese character strokes", *Lecture Notes in Computer Science*, vol. 4285 (*Proceedings of the 21st International Conference on Computer Processing of Oriental Languages (ICCPOL)*), pp. 333-344, December 2006.
[11] Y. T. Tsay and W. H. Tsai. "Attributed String Matching by Split-and-Merge for On-line Chinese Character Recognition", *IEEE Trans. PAMI*, vol. 5, no. 2, pp. 180-185, 1993.
[12] Y. Tonouchi, A. Kawamura., "An On-line Japanese character recognition method using length-based stroke correspondence algorithm", *Proceedings of the Fourth Intl. Conf. on Document Analysis and Recognition (ICDAR)*, vol. 2, pp.633-636, August 1997.
[13] Z. H. Hu, H. Leung and Y. Xu. "Stroke Correspondence Based on Graph Matching for Detecting Stroke Production Errors in Chinese Character Handwriting", *Lecture Notes on Computer Science*, vol. 4810 (*Proceedings of Pacific-Rim Conference on Multimedia (PCM 2007)*), pp. 734-743, December 2007.
[14] C. K. Tan, "An algorithm for online strokes verification of Chinese characters using discrete features", *8th Intl. Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pp. 339–344, August 2002.
[15] J. F. Allen, "Maintaining knowledge about temporal interval", *Communication of the ACM*, vol. 26, no. 11, pp.832-843, 1983.
[16] M. Nabil, A. H. H. Ngu and J. Shepherd, "Picture similarity retrieval using the 2D projection interval representation". *IEEE Trans. Knowledge and data engineering*, vol. 8, no. 4, pp.533-539, 1996.
[17] K. T. Tang, K. K. Li and H. Leung. "A web-based Chinese handwriting education system with automatic feedback and analysis", *Lecture Notes in Computer Science*, vol. 4181, (*Proceedings of the 5th International Conference on Web-based Learning (ICWL 2006)*), pp.176-188, July 2006.
[18] W. H. Tsai, and K. S. Fu, "Error-correcting isomorphisms of attributed relational graphs for pattern analysis", *IEEE Trans. SMC*, vol. 9, no. 12, pp.757-768, 1979.
[19] J. Liu, W. K. Cham, and M. M. Y. Chang. "Online Chinese character recognition using attributed relational graph matching", *IEE Proc, Vis. Image Signal Process.*, vol. 143, no. 2, pp. 125-131, 1996.
[20] V. Hautamaki, T. Kinnunen and P. Franti, "Text-independent speaker recognition using graph matching", *Pattern Recognition Letters*, pp.1427-1432, 2008.
[21] C. Freksa, "Temporal Reasoning Based on Semi-Intervals", Artificial Intelligence, vol. 54, pp.199-227, 1992.

Zhi-Hui HU received her Bachelor degree in computer science from Southwest University of Science and Technology, Mianyang, China, in 2005. Currently, she is working on her joint doctoral degree at University of Science and Technology of China and City university of Hong Kong. Her research interests include pattern recognition and Chinese handwriting education.

Yun Xu received his Ph.D. degree in computer science from University of Science of Technology of China, Hefei, China, in 2002.

He has been working in University of Science of Technology of China since 1999. In 2003, he became Associate Professor in the same institute. He is the member of the network teaching for national level excellent course on parallel computing. His research interests include parallel computing, distributed computing and bioinformatics.

Liu-Sheng Huang received his Master degree in computer science from University of Science of Technology of China, Hefei, China, in 1988.

After his graduation, he joined University of Science of Technology of China as a teacher. In 2005, he became the Deputy Dean of Information College. Later, he became the Vice Director of National High Performance Computing Center of Hefei. He is the director of many national 863 and 963 projects. His research interest includes sensor network and information security.

He has won the second prize of the National Science & Technology Award and the National Teaching Achievement Award of China, and also, the first prize of Advancement Technology Awards of Beijing.

Howard Leung received his Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University in 2003.

He is currently an Assistant Professor in the Department of Computer Science at City University of Hong Kong. His current research interests include intelligent tools for Chinese handwriting education, 3D human motion analysis, and web-based learning technologies.

He is currently a member of the Multimedia Systems & Applications Technical Committee (MSATC) which is a subsidiary organization of the IEEE Circuits and Systems (IEEE-CAS) Society. He is also a member of the IEEE Signal Processing Society. He is the Treasurer of the Hong Kong Web Society and the Treasurer of the ACM Hong Kong Chapter.