An Approach for Generating Square Hmong Characters Using Intelligent Derivation Mechanism

Liping Mo[a], Kaiqing Zhou[a], Changfeng Chen[a], Wei Jiang[a]

[a]College of Information Science and Engineering, Jishou University, Jishou, 416000, Hunan, China

Abstract. In order to reduce the storage capacity of fonts and improve the display quality of characters, this study proposes an approach to generate square Hmong characters using intelligent derivation mechanism. In the proposed approach, the process of generating square Hmong characters is transformed into the process of solving the bounding box of components using binary tree. Thus, various square Hmong characters can be generated by repeatedly performing topological transformation of limited components. A corresponding intelligent derivation model is developed to get square Hmong characters based on the glyph data of objective components obtained by intelligent derivation. Meanwhile, deterministic finite automata is also used to represent the square of square Hmong characters. Testing results of the editing software developed based on this model demonstrate that the square Hmong characters generated by the proposed intelligent derivation mechanism can meet the needs of practical applications.

1. Introduction

Various symbols called ideograms are widely used in East Asian languages, such as Chinese and Japanese. In current operating systems, individual character encoding is typically applied to process these ideographic characters. However, these operating systems are confronted with a space extension issue, which restricts further information processing of these languages while applying the method. The scale of coded character set is difficult to reduce because of the huge storage requirement of character data.

To address the space extension issue, a character-generation method was developed to save the storage space of the character set by extracting and connecting components. This study reviews some of the typical applications of this method for generating Chinese characters. Lai et al. [1] developed an inherent heuristic searching method to generate Chinese character by combining layered radicals. In this method, a character combination expression is first obtained by the hierarchical relationship among radicals. Multiple corresponding characters are then generated based on the obtained expression after the radicals are connected. Finally, the optimal character among the generated characters is identified by implementing the inherent heuristic searching algorithm. Shin et al. [2] presented a novel method for generating personal handwritten Chinese glyph by splicing stroke vectors. Two databases are established in the generation process based on

compressed and cited characters. Personal handwritten characters are created by quantification, compression, and the corresponding operations among the stroke vector data of these two databases. Song et al. [3] improved the description method of the existing Chinese character structure by utilizing the similarity of the figure and presented an algorithm for calculating the similarity of the shape based on the structure description. This method does not require the learning and training of glyph sample instances to calculate the similarity, has good adaptability to calculate the similarity between common words and rare words that are difficult to obtain written samples, and can meet the needs of the growing set of Chinese characters to calculate similarity. Experiments show that the similar words list of 6763 characters in GB2312 calculated by this method is in good consistency with human cognitive results, and is used in computer-aided proofreading systems to modify the prompts of different characters, showing good results. Tan et al. [4] proposed a method for recognizing printed Chinese characters using the position and mosaic relation of radicals. In this method, an affine sparse matrix was used to decompose Chinese characters and automatically extract radicals. Wu et al. [5] put forward an auto-generation technology of the Chinese character glyph based on font library. This technology first maps the Chinese character font j to the font description database, uses the font description database as a bridge to digitize the Chinese character font, and then finds the skeleton information corresponding to the Chinese character from the font description library, extracts the stroke skeleton, and by reverse processing, the stroke skeleton information is restored to various Chinese character forms to form a Chinese character font library. Yang et al. [6] put forward to comprehensive Chinese fonts generating scheme. This scheme first establishes a decomposition database for stroke segmentation and feature extraction, and then uses a search algorithm based on stroke splitting to achieve rapid generation of new Chinese characters through stroke reorganization. The experimental result shows that Chinese characters with a new style can be generated rapidly with the proposed scheme. The usability tests show that the performance from the generated characters is similar to the original characters in style-consistency test. Wang et al. [7] proposed a method for similarity calculation of Chinese character glyph based on triple recursive representation. This algorithm represents the Chinese character as a triple: Chinese character structure, the first part of Chinese character and Chinese character tail parts. The component is an operation object, the character structure is an operator, and the character is described as a prefix expression. By establishing a recursive model for calculating the similarity of Chinese characters, the computational process is decomposed layer by layer into similarities between atomic components, and the computational complexity is effectively reduced. Experimental results show that the proposed method is feasible and effective, and has a high coincidence with human perception. The literature [8] presents an approach to creating Chinese calligraphy with a particular style from learning author's written works automatically. In this method, Chinese character topological is represented by fuzzy relative position, topological features are fed into the evaluation model using a decision tree algorithm. Furthermore, an heuristic algorithm, a hypothesis testing and a decay function of transformation amplitude are used to improve the converging speed. The experiments demonstrate the proposed method can obtain the similar style Chinese calligraphy with training samples. Yamamoto et al. [9] proposed a numerical model of the network of two-Chinese-character compound word. According to this network, a Chinese character is a node, and a two-Chinese-character compound word links two nodes. The basic framework of the model is that an important character gets many edges. The importance of a character is represented by the frequency of its appearing in publication. The network generated by the model is small-world and scale-free and reproduces statistical properties in the actual two-character network quantitatively. Li et al. [10] proposed a handy system to automatically synthesize personal handwritings for all Chinese characters in the font library by learning style from a small number (as few as 1%) of carefully-selected samples written by an ordinary person. Experiments including Turing tests with 69 participants demonstrate that the proposed system generates high-quality synthesis results which are indistinguishable from original handwriting. Using this system, for the first time, the practical handwriting font library in a user's personal style with arbitrarily large numbers of Chinese characters can be generated automatically. In view of the problem of wrongly written character teaching in Chinese language teaching, Li et al. [11] put forward a simple, convenient, and efficient input method of wrongly written characters and realized a dynamic generating-and-editing system for wrongly written Chinese characters font, which solves the problems of real-time editing, coding, and inputting of wrongly written characters in editing process using dynamic editing technology, and provides a convenient

input method of wrongly written characters in editing, printing, typesetting. Li et al. [12-14] realized the dynamic generation of point strokes by defining weight vectors of feature points, interpolates sparse data described by pen elements, and achieved special glyph output by defining basic pen elements and extending stroke elements. The literature [14] presents a generation model of Chinese character based on structure and style. The model is described by the stroke element, stroke element vector, path vector, string vector and yoke vector. According to this method, fonts that can be used in the design of True type personalized Chinese character fonts are generated dynamically, and the Web storage and output of the Chinese character fonts on the client are realized. This model provides an effective strategy and method to solve the cloud storage and cloud font services for personalized Chinese characters information. Xiong et al. [15] proposed a new method for automatically generating the Web fonts of Chinese characters. This method uses the dynamic description library to record the features of Chinese characters, and utilizes operations such as stroke extraction, connection, and optimization of Chinese characters based on feature points descriptions, to generate the Web fonts that meet user requirements on the web page. Liu et al. [16] put forward an innovative method to align two glyph contours with three steps. Firstly, 2D Bezier curve control points of glyph contours of each character are expanded into 3D space. Secondly, a Gaussian Mixture Model (GMM) is constructed using this 3D point set. Finally, alignment is established by minimizing Euclidean Distance between two GMMs, and then a transformation is applied accordingly. Experiments results verify the feasibility and effectiveness of the proposed method.

Square Hmong characters are ideograms with a fixed structure, are commonly used in the daily lives of Hmong people in the Wuling mountain area of China. Compared with other ideograms, information processing on square Hmong characters has not been extensively studied. Code points of square Hmong characters have not been allocated in Unicode and other character sets.

According to [17], the word-information principles of square Hmong characters can be summarized in two aspects.

(1) Using some simple Chinese characters, Chinese radicals and symbols without pronunciation and meaning (such as ~ and X) to represent the components of phonetic-symbol, meaning-symbol, or shape-symbol.

(2) Using 'One word and one syllable' method to mark a morpheme or word.

The structure of square Hmong characters can be divided into four types, namely, left-right structure, top-bottom, part-enclosed, and internal-external structures. In particular, the part-enclosed structure can be further classified into three types, namely, upper-left-enclosed, lower-left-enclosed, and upper-right-enclosed types. Some typical examples of square Hmong characters with different structures are listed in Fig.1.

Mark	Structure Type	Example	Meaning
w_1	left-right structure	吋	call names
w_2	left-right structure	嫐	poor
w_3	left-right structure (3 components)	抍	dog
w_4	top-bottom structure	尖	beggar
w_5	top-bottom structure	虫	snake
w_6	top-bottom structure (3 components)	畠	nap
w_7	upper-left-enclosed structure	痈	wound
w_8	lower-left-enclosed structure	冠	comb
w_9	upper-right-enclosed structure	氃	fly
w_{10}	internal-external structure	闽	go outside

Fig. 1. Examples of square Hmong characters with different structures.

According to the word-information principle, if a square Hmong character consists of three or more components, two or three of these components can be combined into simple Chinese. The simple Chinese generated would be considered a component in this square Hmong character. The results of statistical analysis of the 1,127 square Hmong characters collected indicate that most of the Hmong characters are organized by two components, whereas only a few Hmong characters with left-right structure and top-bottom structures are organized by three components.

In previous work, Mo et al. [18] proposed a dynamic approach for generating square Hmong characters using the combinatorial expression of components. With the OpenType font technique, the proposed approach can generate various Hmong characters with different structures that do not occupy Unicode encode points. However, this method is based on glyph data, cannot display character whose glyph data is not available.

To address this limitation, a novel approach for generating square Hmong characters using intelligent derivation mechanism is presented. In this new approach, a large number of square Hmong characters can be generated by topological transformation of finite components.

The rest of this study is organized as follows. Section 2 introduces the method for representing square Hmong characters. Section 3 introduces related concepts and operations of the intelligent derivation of square Hmong characters. Section 4 presents an intelligent derivation mechanism of square Hmong characters. Section 5 illustrates a case study of the proposed intelligent derivation function of square Hmong characters. Finally, Section 6 concludes the paper.

2. Representation of square Hmong characters

Table 1 presents the 16 types of components of square Hmong characters.

Table 1: Types of square Hmong characters and corresponding marks.

Number	Name	Mark
1	left component	c_l
2	right component	c_r
3	top component	c_t
4	bottom component	c_b
5	upper-left-external component	c_{ule}
6	lower-right-internal component	c_{lri}
7	lower-left-external component	c_{lle}
8	upper-right-internal component	c_{uri}
9	upper-right-external component	c_{ure}
10	lower-left-internal component	c_{lli}
11	external component	c_e
12	internal component	c_i
13	right-left component	c_{rl}
14	right-right component	c_{rr}
15	bottom-left component	c_{bl}
16	bottom-right component	c_{br}

Assuming that SG and C_i(i=start, end, first, middle, final) are used to represent square Hmong characters and their components, two- and three-component square Hmong characters can be illustrated as $SG=\{C_{start}C_{end}\}$ and $SG=\{C_{first}C_{middle}C_{final}\}$, respectively, where $C_{start}=\{c_l, c_t, c_ule, c_{lle}, c_{ure}, c_e\}$, $C_{end}=\{c_r, c_b, c_{lri}, c_{uri}, c_{lli}, c_i\}$, $C_{first}=\{c_l, c_t\}$, $C_{middle}=\{c_{rl}, c_{bl}\}$, and $C_{final}=\{c_{rr}, c_{br}\}$.

Deterministic finite automata (DFA) representation of square Hmong characters based on the 16 types of components is proposed as follows.

Definition 1. DFA representation of square Hmong characters: The DFA can be expressed as $M = (K, A, F, S, Z)$, where,

(1) K is a finite set of states, $K=\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$;

(2) A is a finite set of input symbols, $A=\{c_l, c_t, c_{ule}, c_{lle}, c_{ure}, c_e, c_r, c_b, c_{lri}, c_{uri}, c_{lli}, c_i, c_{rl}, c_{bl}, c_{rr}, c_{br}\}$;

(3) F is a map set on $K \times A \rightarrow K$, where $F=\{f(0,c_l)=1, f(0,c_t)=2, f(0,c_{ule})=3, f(0, c_{lle})=4, f(0,c_{ure})=5, f(0,c_e)=6, f(1,c_{rl})=7, f(1,c_r)=9, f(2,c_{bl})=8, f(2,c_b)=9, f(3,c_{lri})=9, f(4,c_{ri})=9, f(5,c_{lli})=9, f(6,c_i)=9, f(7,c_{rr})=9, f(8,c_{br})=9\}$;

(4) $S \in K$ is the initial or start state, and $S=0$;

(5) $Z \subset K$ is a set of accepted states, and $Z=\{9\}$.

Fig.2 presents the state graph of DFA, where in any square Hmong character can be generated by a given path from the initial state 0 to the accepted state 9.
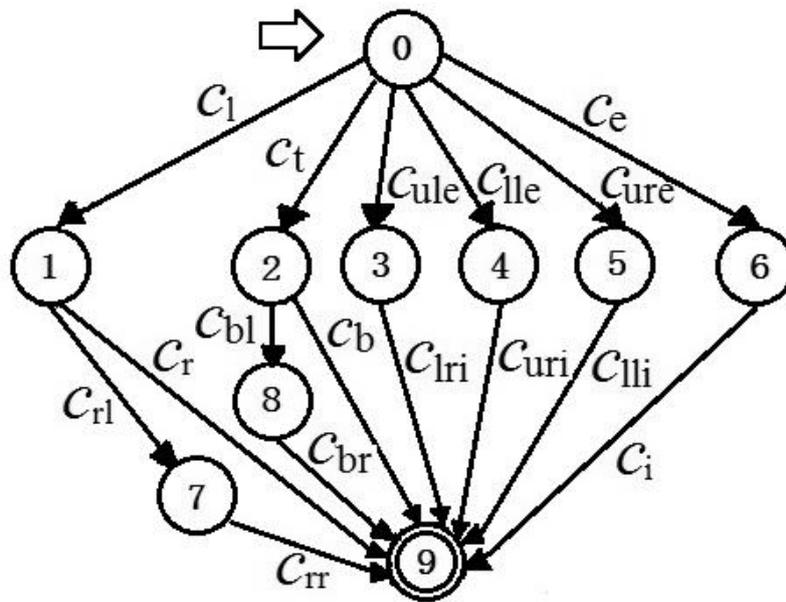

Fig. 2. State graph to generate square Hmong characters.

3. Related concepts and operations of intelligent derivation of square Hmong characters

Two important concepts in this study, namely, original component and objective component, are defined as follows.

Definition 2. original component and objective component: Chinese characters, Chinese radicals, and pure symbols are considered as original components of Hmong characters. Objective component is defined as a constituent part while displaying square Hmong characters.

The objective component can be obtained by topological transformation of the original component. Thus, a square Hmong character can be generated while all of the corresponding objective components are obtained by topological transformation of the original components.

3.1. Definition and normalization of the component bounding box

The bounding box is a simple geometric graph that reflects the position and size of complex geometric objects. To simplify the transformation, the image processing of the Hmong character component is converted into the processing of bounding box. The bounding box of square Hmong character component is defined as follows.

Definition 3. Bounding box of square Hmong character component: The corresponding width and height
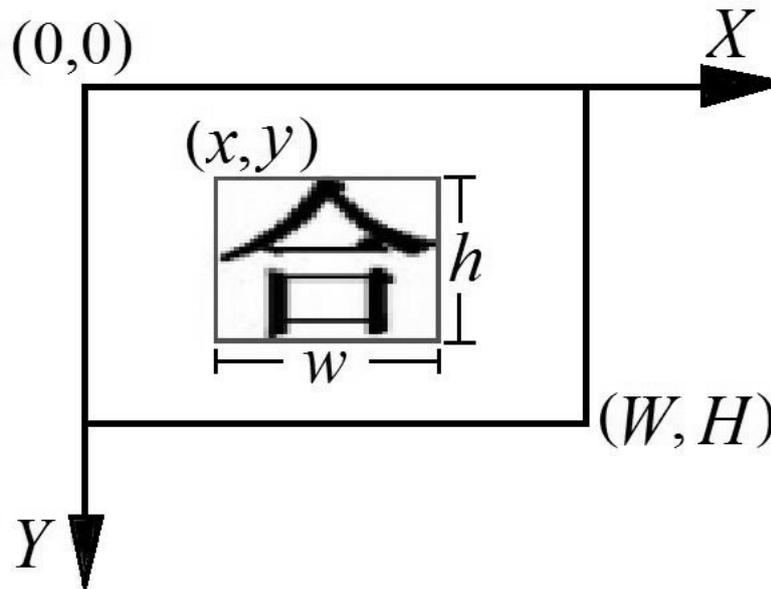
Fig. 3. An example of a bounding box of square Hmong character component.

for a given square Hmong character grid are W and H, respectively. The Cartesian coordinate system is established with the origin at the upper-left corner of the grid, wherein the X axis is positive in the right direction, and the Y axis is positive in the downward direction. In this coordinate system, the bounding box of square Hmong character component is a minimum enclosing rectangle determined by upper-left vertex coordinates (x, y), width (W), and height (H) of the image of a square Hmong character component. The bounding box is further represented as a four-tuple $B = (x, y, w, h)$ where $x \in (0, W)$, $y \in (0, H)$, $w \in (0, W - x)$, and $h \in (0, H - y)$. Fig.3 illustrates an example of a bounding box of square Hmong character component.

The bounding box is normalized by $\bar{B} = (\frac{x}{W}, \frac{y}{H}, \frac{w}{W}, \frac{h}{H})$ based on the topological invariance. The bounding boxes mentioned in the rest of this study are all processed by normalization.

3.2. Determination of topological transformation coefficients of the bounding box

The generation process of square Hmong characters can be converted into a calculation process of the corresponding bounding box of the components. This calculation process can be considered as a process to gradually generate the bounding box of the objective component from the standard bounding box of the original component by calculating the topological transformation coefficients.

For two given bounding boxes $B = (x, y, w, h)$ and $B' = (x', y', w', h')$, the topological transformation coefficients from B_1 to B_2 are obtained based on the following equation

$$\begin{cases} R_w = w'/w \\ R_h = h'/h \\ x = x + x.R_w \\ y = y + y.R_h \end{cases} \tag{1}$$

In actual operations, the relevant information of the standard bounding box of the original component can be directly gathered from the corresponding Chinese character fonts. Thus, the determination issue of topological transformation coefficients of the bounding box is observed in the generation of the bounding box of the objective component.

3.3. Transformation operations of the bounding box of the component

The purpose of solving the bounding box is to calculate the position and size of the bounding box of the objective component, which involves transformation operations, such as centering, tiling, and aspect ratio adjustment.

3.3.1. Centering

Centering is used to adjust a bounding box to the middle position of another bounding box based on a given direction of a coordinate axis. In this operation, the width and height of the bounding box will remain consistent, whereas the upper-left vertex coordinate will change.

For two given bounding boxes $B_1 = (x_1, y_1, w_1, h_1)$ and $B_2 = (x_2, y_2, w_2, h_2)$, centering is used to move B_1 into B_2 along three different directions (X-axis direction, Y-axis direction, and simultaneously in the X- and Y-axis directions). Therefore, the new bounding box $B_1 = (x, y, w_1, h_1)$ under these three cases is generated based on Eqs.(2) to (4), respectively.

$$\begin{cases} x = x_2 + |w_1 - w_2|/2 \\ y = y_1 \end{cases} \tag{2}$$

$$\begin{cases} x = x_1 \\ y = y_2 + |h_1 - h_2|/2 \end{cases} \tag{3}$$

$$\begin{cases} x = x_2 + |w_1 - w_2|/2 \\ y = y_2 + |h_1 - h_2|/2 \end{cases} \tag{4}$$

3.3.2. Tiling

Tiling is used to expand or compress a bounding box into another bounding box along with a given coordinate-axis direction. The new bounding boxes are marked as $B' = (x_2, y_1, w_2, h_1)$, $B' = (x_1, y_2, w_1, h_2)$, and $B' = (x_2, y_2, w_2, h_2)$, which can be generated after tiling along X-axis direction, Y-axis direction, and simultaneously in the X- and Y-axis directions, respectively.

3.3.3. Aspect ratio adjustment

Aspect ratio is an important parameter that reflects the shape of the component in a square Hmong character. Each type of component has its own method of aspect ratio adjustment.

Assuming that a bounding box of the original component is $B = (x, y, w, h)$, a new bounding box marked as B' can be obtained by adjusting aspect ratio r. Depending on the type of component, the aspect ratio adjustment can be divided into the following three cases.

For a left-right structure component, the new bounding box B′ is generated by Eq.(5) when maintaining the same height.

For a top-bottom structure component, the new bounding box B′ is generated by Eq.(6) when maintaining the same width.

For a part-enclosed structure or internal-external structure component, the new bounding box B' is generated by changing the height and width. When $h/w \leq r$, the adjustment operation is implemented by Eq.(5). Otherwise, the adjustment operation is implemented by Eq.(6).

Eq.(5) and Eq.(6) are presented below.

$$B' = (x, y, h/r, h) \tag{5}$$

$$B' = (x, y, w, w * r) \tag{6}$$

4. Intelligent derivation of square Hmong characters and related algorithms

The key point of solving the bounding box of the objective component is to gather the topological transformation coefficients and the position and size of the bounding box. In this study, the derivation process of generating square Hmong characters is executed by a binary tree approach with the "type + component" form. In this binary tree, non-terminal and terminal nodes are used to represent the type of square Hmong characters and the concrete component, respectively. The problem of each node data can be addressed via the decomposed simple subtree using the recursion of the binary tree.

4.1. Design of encoding tree

The encoding tree is a symbolic representation of the derivation tree. In the encoding tree, six types of square Hmong characters are marked by the corresponding letters, as shown in Table 2. The component is represented by the code point of the original component character in the Unicode character set.

Table 2: Types of square Hmong characters and corresponding marks.

Type of square Hmong character	Mark
left-right structure	Z
top-bottom structure	S
internal-external structure	N
upper-left-enclosed structure	A
lower-left-enclosed structure	B
upper-right-enclosed structure	C

The encoding tree of a square Hmong character with two components is considered as a first-level structure with two layers. The encoding tree of a square Hmong character with three components is considered a second-level structure with three layers. According to the encoding tree, the examples w_6 and w_8 in Fig.1 can be encoded as "S 5408 Z 76EE 76EE"and "B 51A0 5B50", respectively. Fig.4(a) and Fig.4(b) illustrate the corresponding derivation trees and encoding trees, respectively.
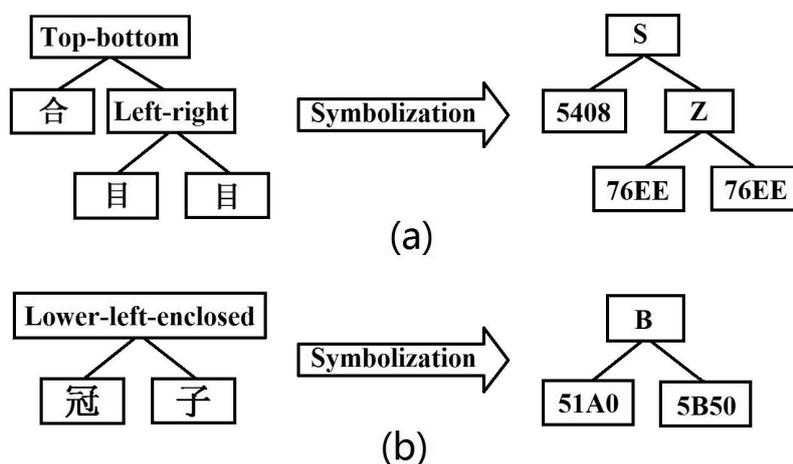

Fig. 4. An example of a bounding box of square Hmong character component.

4.2. Structure design of intelligent derivation mechanism

The intelligent derivation mechanism of square Hmong characters can be separated into three layers, namely, top level, middle level, and bottom level. The functions of the layers from top-to-bottom order are

to decompose the encoding tree, perform structure derivation for square Hmong characters with different structures to solve the topological transformation coefficient of each node and adjust the aspect ratio of components. Given the similarity with the derivation process, the middle level can be further classified into two subsections. One subsection, namely, the derivation module of a square Hmong character with a vertical-horizontal structure, is employed to derive the left-right and top-bottom structures. The other subsection, namely, the derivation module of a square Hmong character with containment structure, is employed to derive the part-enclosed and internal-external structures. The structure of the intelligent derivation mechanism is illustrated in Fig.5.
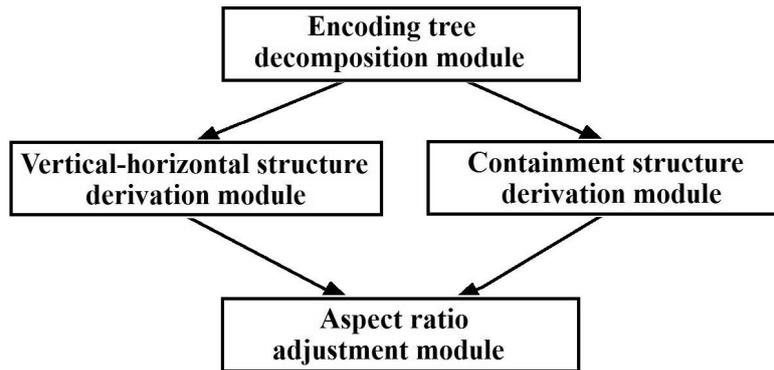

Fig. 5. Structure of the intelligent derivation mechanism.

4.3. Encoding tree decomposition module

The process of decomposing the encoding tree consists of two phases. In the first phase, the tree is decomposed into a series of subtrees by implementing a depth-first-based pre-order traversal. In the second phase, two operations will be performed repeatedly until all subtrees are settled. The homologous structure derivation module will be used to execute for each decomposed subtree to achieve the goal solution. And the solution is used to substitute for the original subtree.

According to this process, Algorithm 1 is proposed to achieve the function of the encoding tree decomposition module below.

Algorithm 1. Encoding tree decomposition algorithm

Input: Unicode encoding sequence of the components and the structure type of a square Hmong character

Output: Transformation parameters of the objective components

Steps:

Step 1: The encoding tree for a square Hmong character is generated by combining the Unicode encoding sequence of the components and the structure types.

Step 2: The encoding sequence of a square Hmong character is obtained by implementing a depth-first-based pre-order traversal on the encoding tree.

Step 3: The encoding tree is decomposed into a series of subtrees based on the obtained encoding sequence.

Step 4: The transformation parameters of all original components of a square Hmong character are generated by calling the corresponding structure derivation module.

Step 5: The objective components are obtained by performing topological transformation using these parameters.

4.4. Derivation module of a square Hmong character with a vertical-horizontal structure

The shape of the component shown in a square Hmong character with a vertical-horizontal structure is determined by two factors, namely, the order of appearance in the square Hmong character and the size of

this component. The proportion of the component in the square Hmong character is directly proportional to the size of the original component. Thus, the bounding box of the component of a two-component square Hmong character is directly achieved from the standard bounding box of the original component. The bounding box of the component of a three-component square Hmong character is obtained from the depth-first traversal result of the spanning tree.

All related information about the bounding box of each component is stored in the component-position information database. This database contains a series of data, such as component name, component type, structure name, position number, and the list of the maximum bounding box of each position. The maximum bounding box of each component can be obtained via the corresponding transformation operations of related data extracted from the above database. The derivation function for a square Hmong character with a vertical-horizontal structure can be achieved using Algorithm 2.

Algorithm 2. Derivation algorithm of a square Hmong character with a vertical-horizontal structure
    Input: Related data of components stored in the component-position information database
    Output: Transformation coefficients of each component
    Steps:
    Step 1: The proportion of each component in a square Hmong character is calculated according to the related data of components stored in the component-position information database.
    Step 2: The maximum bounding box is obtained based on the order of each component in a square Hmong character.
    Step 3: For a square Hmong character with a left-right structure, each component in the maximum bounding box of this character is centered along the *X*-axis direction using Eq.(2). For a square Hmong character with a top-bottom structure, each component in the maximum bounding box of this character is centered along the *Y*-axis direction using Eq.(3).
    Step 4: The corresponding bounding box of all components of a square Hmong character is generated after adjusting the aspect ratio of each component by calling Algorithm 4.
    Step 5: All transformation coefficients of each component are calculated using Eq.(1).

4.5. Derivation module of a square Hmong character with a containment structure

The common highlight of a square Hmong character with an internal-external structure is that the external component acts as a framework, and the interior component is embedded in this framework. The maximum bounding box of each component will be determined by the external component. All of the related information data of each component is also stored in the component-position information database. The maximum bounding box of each component can be obtained via the corresponding transformation operations of related data extracted from the above database. The derivation function of a square Hmong character with a containment structure can be achieved based on Algorithm 3.

Algorithm 3. Derivation algorithm of a square Hmong character with a containment structure
    Input: Related data of components stored in the component-position information database
    Output: All of the transformation coefficients of each component
    Steps:
    Step 1: According to the related data of components stored in the component-position information database, the maximum bounding box of the external component is marked as the actual bounding box of the character.
    Step 2: The maximum bounding box of the interior component is obtained after adjusting the size of the standard bounding box of the interior component by calling Algorithm 4.
    Step 3: The obtained bounding box is simultaneously tiled to the maximum bounding box of the character along the *X*- and *Y*-axis directions.
    Step 4: All transformation coefficients of each component are calculated using Eq.(1).

4.6. Aspect ratio adjustment module

In the intelligent derivation procedure, each structure derivation function needs to call the function of aspect ratio adjustment. Based on the related data of the components from the aspect ratio database,

components can be adjusted using different adjustment approaches. The aspect ratio database includes a group of items, such as component name, component type, structure name, number of occurrences, number of components, and related data of aspect ratio (such as minimum, maximum, mean, and variance). The function of aspect ratio adjustment can be achieved using Algorithm 4.

Algorithm 4. Aspect ratio adjustment algorithm

   Input: Related data of components stored in the aspect ratio database

   Output: Adjusted aspect ratio

   Steps:

   Step 1: The type of the component is determined according to the related data of components stored in the aspect ratio database. If the component is that of the square Hmong character with a vertical-horizontal structure, move to Step 2. Otherwise, move to Step 3.

   Step 2: The related data of aspect ratio is determined according to the related data of components stored in the aspect ratio database. For a square Hmong character with left-right structure, if the current aspect ratio of the component is between the minimum and the maximum, it remains unchanged; otherwise, it is adjusted to the average using Eq.(5). For a square Hmong character with a top-bottom structure, if the current aspect ratio of the component is between the minimum and the maximum, it remains unchanged; otherwise, it is adjusted to the average using Eq.(6).

   Step 3: The related data of aspect ratio is determined according to the related data of components stored in the aspect ratio database. For a square Hmong character with a containment structure, if the current aspect ratio of the component is equal to the average, it remains unchanged; if the current aspect ratio is higher than the average, it is adjust to the average using Eq.(6); otherwise, it is adjusted to the average using Eq.(5).

5. Case study

This section presents a case study to verify the feasibility of the proposed intelligent derivation mechanism of square Hmong characters.

5.1. Example of the Intelligent Derivation of Square Hmong Characters

According to Algorithms 1 to 4, the processes of generating a square Hmong character can be separated into a series of solution processes of the transformation coefficients of components. And these components belong to a Hmong character with a first-level structure from the leaf node of the encoding tree.
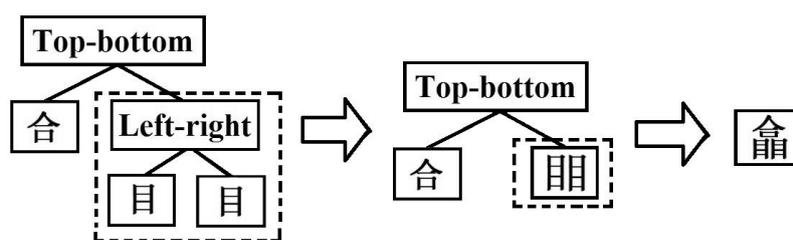

Fig. 6. Intelligent derivation process of w_6.

A typical square Hmong character, which is marked as w_6 in Fig.1, is used to illustrate the intelligent derivation process of generating a square Hmong character. Fig.6 depicts the intelligent derivation process of w_6. According to Fig.6, for the character w_6, the solution process of transformation coefficients of component can be described in the following steps.

   Step 1: According to Fig.4(a), the character w_6 has a second-level structure, and the top structure and final structure of w_6 are S and Z, respectively.

   Step 2: The final structure is separated from w_6, and is considered as a pseudo-square Hmong character to obtain the generation encoding sequence. As shown in Fig.7(a), the generation encoding sequence

is "Z 76EE 76EE". After implementing the derivation module of the square Hmong character with a vertical-horizontal structure, the bounding boxes of two sub-components, namely, "76EE" and "76EE", are generated in the pseudo square Hmong character.

Step 3: The pseudo square Hmong character (marked as "#1") acts as a component, and is merged with "#1" into the upper structure. As shown in Fig.7(b), the current generation encoding sequence is "S 5408 #1". After implementing the derivation module of the square Hmong character with a vertical-horizontal structure, the bounding boxes of two new sub-components, namely, "5408" and "#1", are generated as the bounding boxes of objective components.
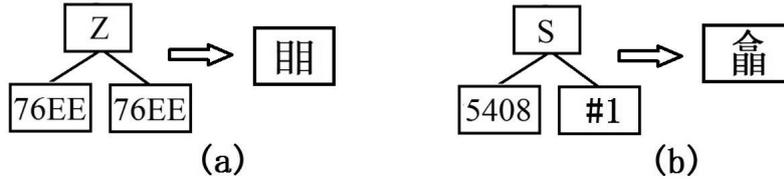

Fig. 7. Derivation result of w_6.

Step 4: All the transformation coefficients of each component of w_6 are calculated using Eq.(1).

Step 5: The corresponding objective components are obtained by executing the topological transformation according to the above coefficients.

Step 6: The character w_6 is generated by combining all of the objective components.

5.2. Application of the proposed intelligent derivation mechanism

A font library of objective components is created based on the data generated using intelligent derivation mechanism. Some glyphs of objective components in the font library are adjusted properly to ensure the accuracy of the component image of a square Hmong character. The font library is used to input characters in test experiments. The testing results are demonstrated in Fig.8.
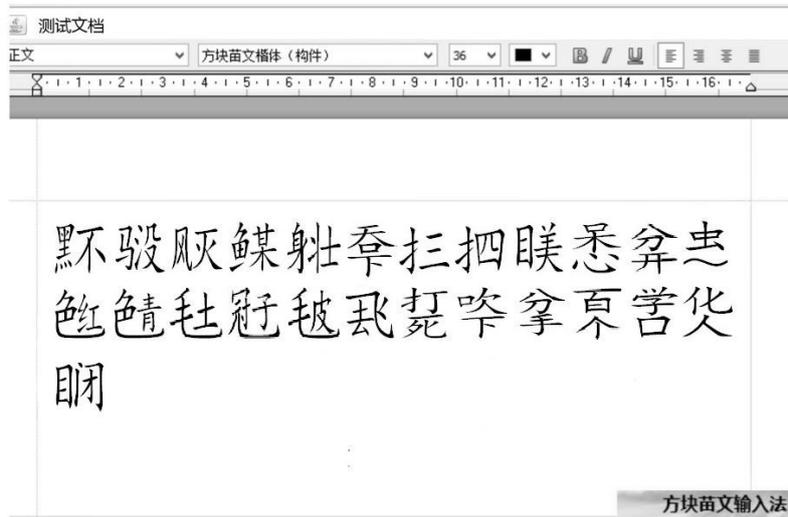

Fig. 8. Input testing result of square Hmong characters.

As shown in Fig.8, the generated square Hmong characters have met the demands of the practical applications.

6. Conclusion and future work

The proposed intelligent derivation mechanism of square Hmong characters has three characteristics. First, the component transformation is the core of this intelligent derivation mechanism. The transformation coefficients are achieved by calculating the bounding boxes of components. Second, the binary tree is used to recursively calculate the transformation coefficients in this mechanism. This characteristic is an important aspect to reflect the rationality of the character derivation, which also ensures the simplicity and validity of this mechanism. Third, every intelligent derivation function is designed independently, and the data needed for intelligent derivation is stored in the separate information database. This approach facilitates the adjustment and optimization of the derivation process in the future.

In our future work, we will develop a complete database of the information data for the component transformation based on the 1,127 square Hmong characters collected so far.

References

[1] P.K. Lai, D.Y. Yeung, M.C. Pong, A heuristic search approach to Chinese glyph generation using hierarchical character composition, Computer processing of oriental languages, 10 (1996), 307-323.
[2] J. Shin, K. Suzuki, A. Hasegawa, Handwritten Chinese character font generation based on stroke correspondence, International journal of computer processing of oriental languages, 18 (2005), 211-226.
[3] R. Song, M. Lin, S.L. Ge, Similarity calculation of Chinese character glyph and its application in computer aided proofreading system, Journal of Chinese computer systems, 29 (2008), 1964-1968.
[4] J. Tan, X. Xie, W.S. Zheng, et al, Radical extraction using affine sparse matrix factorization for printed Chinese characters recognition, International journal of pattern recognition and artificial intelligence, 26 (2012), 1250005-1250024.
[5] Q.X. Wu, Q.S. Li, Research of Chinese character auto-generation technology based on the dynamic description library, Science technology and engineering, 8 (2012), 28-33.
[6] G. Yang, H. Liang, Y. Su, Generating Chinese characters based on stroke splitting and feature extraction, Displays, 34 (2013), 258-269.
[7] D. Wang, S.H. Xiong, New algorithm for similarity calculation of Chinese character glyph, Application research of computers 30 (2013), 2395-2397.
[8] W. Li, C. Zhou, Automatic creation of artistic Chinese calligraphy, Journal of Software, 8 (2013), 3049.
[9] K. Yamamoto, Y. Yamazaki, Structure and modeling of the network of two-Chinese-character compound words in the Japanese language, Statistical mechanics and its applications, 412 (2014), 84-91.
[10] W. Li, Y. Song, C. Zhou, Computationally evaluating and synthesizing Chinese calligraphy, Neuro computing, 135 (2014), 299-305.
[11] X. Li, Q.S. Li, Dynamic generation and editing system for wrongly written Chinese characters font, International journal of multimedia and ubiquitous engineering, 10 (2015), 31-40.
[12] Q.S. Li, J. Xiong, Q.X Wu, et al, Study of feature weighted-based generation method for dian strokes of Chinese character, Acta scicentiarum naturalum universitis Pekinesis, 50 (2014),153-160 (in Chinese).
[13] Q.S. Li, L. Zhang, Q. Liu, et al, Chinese character computing model based on cloud information protection, Computer science, 42 (2015),73-79.
[14] J. Xiong, Q.S. Li, F. Gao, et al, Generation method of Chinese Webfont based on dynamic font description library, Computer applications and software, 33 (2016), 61-65 (in Chinese).
[15] Y.B. Liu, Y.N. Sun, E.D. Xun, Chinese calligraphy alignment based on 3D point set registration, Acta scicentiarum naturalum universitis Pekinesis, 52 (2016), 81-88 (in Chinese).
[16] Q.S. Li, Q. Xu, J.G. Xiao, et al, A structure and style model for Chinese character dynamic generation, Acta scicentiarum naturalum universitis Pekinesis, 53 (2017), 219-229 (in Chinese).
[17] Z.B. Yang, H.Y. Luo, On the Folk Coinage of Characters of the Miao People in Xiangxi Area, Journal of Jishou university (social sciences edition), 29 (2008), 130-134 (in Chinese).
[18] L.P. Mo, K.Q. Zhou, A dynamical glyph method of Xiangxi folk Hmong characters and its implementation approach, Acta scicentiarum naturalum universitis Pekinesis,52 (2016),141-134 (in Chinese).