Decomposition Chinese Character into Components Based on Compactness and HOP

Hu Xin
Nanchang Institute of Technology, Nanchang, Jiangxi, 330099, China, 2471152822@qq.com

Abstract: Decompose Chinese Character into Components (DCCC) is a boring and waste time and energy job. A method utilizing image process technique is proposed to DCCC automatically. Characters are segmented into pieces with connectivity firstly. Then the parts are the matched with components images and re-segmented. Finally the components found are clustered into large components. Experiments show that our method has high accuracy and bearable speed.

Keywords: Chinese Character Component; Image segmenting; Histogram of Projection; Clustering.

I. INTRODUCTION

Nowadays, touch screen is widely used in Haptic Devices, Smart Phone, Tablet PC, e-reader, and exhibition set, which benefit from the development of Hand Writing Recognition (HWR). But HWR is not an absolute mature technique yet. Though recognition rate surpass 90% is made on most popular sets, Misrecognition and multiple-choice makes one mad sometimes. So many efforts are still being spent on research of HWR. As the complex structure it is, Chinese Character is more difficult, and Handwriting Chinese Character Recognition (HWCCR) is always a big challenge. Among the known methods, many HWCCR algorithms need to decompose the character into components firstly.

A project to Decompose Chinese Character into Components (DCCC) was finished manually years ago, organized by Chinese Government, with Lots of manpower and materials. But unfortunately, only Simplified Chinese Characters are considered in this project. As communication between China, Taiwan, Hong Kong, Japanese, and Korea increases, more and more Traditional Chinese and Kanji are imported to Chinese Characters, so that the old DCCC result can’t fit the new situation any more. DCCC with manual ways is not only a time-consuming work but also a job with high error rate, which means we must find a way to decompose character automatically.

Chinese characters are recorded as point matrices in computer, which can also be viewed as images. To decompose a character into components is then a problem about image segmentation. But it’s not a simple segmentation by colors or edges, as segment results should match the standard components images. To obtain a full components database with less errors, we tried a method include three steps, connective area detecting and segmenting, area matching, area re- segmenting, and area clustering.

II. RELATED WORK

Image segmentation is a conventional problem in image processing field. Many excellent algorithm for image segmentation had been proposed in past years. Among them the algorithm named Mean-Shift [1] is most soundly and focused, which needs color images support, while only black and white colors are included in font images commonly. The more suitable method for character dividing is threshold segmentation[2][3], but the purpose of DCCC is not only to divide characters into pieces but also to cast them to special components, so more works than threshold segmentation should be done.

Because the importance to Chinese character structural analysis, Long-term attention has been paid to DCCC, Chinese Character Component (CCC) has different definition to each researchers, PAN Defu and ZHAN Zhenquan[4] defined CCC with separability and Independence, and proposed eight principles to DCCC, listed as bellow: Multi-Stroke has priority; Do not split intersection; split all skims appended on other components; Use the across of Character “Ge”, in both sides when it was connected to others; Don’t Split a circle; Don’t split a cross structure; Don’t split a vertical and lines upper or below it; Don’t split a point and its bound. PAN’s work can be utilized in DCCC program as splitting principles. Different font has different component shapes, Zhao Wei[5] et al. built up a set of handwritten components, and proved that good components can promote recognition rate of handwritten characters.

III. CONNECTIVE SEGMENTING

Connectivity is one of the most remarkable features of CCC and a possible way to gain components. A component is often a connective area, and a connective part of Chinese character is often a component too. So it is helpful for DCCC to divide characters into connective parts. Connective Segmentation is a traditional problem in image processing field, which classify images into parts decrypted as form(1.1):

\[ P_i = \{ p | p \notin P_i, C(p) > 0, \exists q \in P_i, \text{dist}(p, q) = 1 \} \]

\[ \cup P_i = \{ p | p \in I, C(p) > 0 \} \]

(1.1)
In form (1.1), letter \( I \) expresses the processed Image, and \( C(p) \) denotes color at \( p \) of \( I \).

Connective Segmentation is often implemented by a recursive method, which consumes much RAM and CPU resource. To accelerate this procedure, a chain with two pointers was designed to change Connective Segmentation into single loop method. The procedure of Connective Segmentation by chain with two pointers is shown in figure 1.

After Connective Segmentation, images are divided into pixel sets, which are not components necessary. Some segmentation results are shown in figure 2.

After components matching and re-segmenting, all character pixels are classified into different components, but this is not the end. While a component could contain other components, components divided by connectivity

**IV. COMPONENTS MATCHING AND RE-SEGMENTING**

To compare components, a compound descriptor for components should be built up. First of all, we found that the rate of black pixels of each component is relative stable in its minimum bounding rectangle. We call the rate of black pixels of a component as Compactness, which is defined as formula (1.2).

\[
\text{Compactness} = \frac{\text{count(black rect)}}{\text{width}_{\text{rect}} \times \text{height}_{\text{rect}}}
\]

\[\forall p \in \text{component}, p \in \text{rect}\]

\[\exists p \in \text{component}, p \cdot x = \text{rect} \cdot \text{left}, s.t. p = \text{black}\]

\[\exists p \in \text{component}, p \cdot y = \text{rect} \cdot \text{top}, s.t. p = \text{black}\]

\[\exists p \in \text{component}, p \cdot x = \text{rect} \cdot \text{right}, s.t. p = \text{black}\]

\[\exists p \in \text{component}, p \cdot y = \text{rect} \cdot \text{bottom}, s.t. p = \text{black}\]

\[(1.2)\]

Compactness is a simple and accuracy feature to recognize components, but it doesn’t work well when distortion exists. More features should be employed to distinguish components further. Histogram of Projection (HOP), defined as formula (1.3), is a simple statistic feature kept geometry information.

In formula (1.3), each \( \text{HOP}_H \) and \( \text{HOP}_V \) represent HOP in horizontal direction and vertical direction. When distortion exists in the component image, HOP will be translated and scaled, and a simple comparison can’t work normally. The solution of this problem is comparing histograms by correlation method, which can be described by formula (1.4).

\[
dist(\text{HOP}_1, \text{HOP}_2) = \frac{\sum_{\text{HOP}_x = \text{HOP}_2} \text{HOP}_x \cdot \text{HOP}_y}{\sum_{\text{HOP}_x = \text{HOP}_2} \text{HOP}_x^2 \cdot \sum_{\text{HOP}_y = \text{HOP}_2} \text{HOP}_y^2}
\]

\[(1.4)\]

Formula (1.4) describes the full relationship between \( \text{HOP}_1 \) and \( \text{HOP}_2 \), but the directivity of components would be ignored, which makes some components indistinguishable. We simplify the correlation method to distinguish direct of HOP by computing dist sequentially, expressed as formula (1.5).

\[
dist(\text{HOP}_1, \text{HOP}_2) = \frac{\sum_{\text{HOP}_x = \text{HOP}_2} \text{HOP}_x \cdot \text{HOP}_y}{\sum_{\text{HOP}_x = \text{HOP}_2} \text{HOP}_x^2 \cdot \sum_{\text{HOP}_y = \text{HOP}_2} \text{HOP}_y^2}
\]

\[(1.5)\]

Not all the parts divided by connective are single components. Matches of HOP or Compactness would be failed when more than one component exist in a pixels set. Re-segmenting should be performed when matches failed. According PAN’s[4] work, the connected components have only three position types between them, containing, horizontal, and vertical. A recursive algorithm is designed to re-segment pixel sets into components.

Algorithm 1: (Pixels sets re-segmenting algorithm)

**Step 1**: Let divideRate=0.1, divideType=0, bestMatch1=0, bestMatch2=0.

**Step 2**: If divideRate>=1, let divideType=divide Type+1, divideRate=0.1; Divide pixels set \( S \) into two parts, with divideRate and 1-divideRate heights of whole set, switch divideType, case 0 in up and down direction, case 1 in left and right direction, otherwise in inner and exterior direction.

**Step 3**: Find the best matches of the two parts, get their match degrees matchDegreel and matchDegre2. If matchDegreel>bestMatch1, let bestMatch1=match Degree1. If matchDegre2 > bestMatch2, let bestMatch2=match Degree2. If bestMatch1+ bestMatch2 > threshold, end, else if bestMatch1> threshold1, re-segment part2, end; else if bestMatch2>threshold1, re-segment part1, end; else let divideRate=divideRate+0.1, goto step 2.

**V. COMPONENTS CLUSTERING**

After components matching and re-segmenting, all character pixels are classified into different components, but this is not the end. While a component could contain other components, components divided by connectivity
could be parts of other components. A hierarchical algorithm is imported to form components, where distance of components is defined as below, in formula (1.6).

\[
dist(C_1, C_2) = 1 - \frac{\text{count}[C | C_1 \subseteq C, C_2 \subseteq C]}{\text{count}[C | C_1 \subseteq C] + \text{count}[C | C_2 \subseteq C]} 
\] (1.6)

With this definition of distance, the main step of clustering algorithm is clustering the two clusters with minimum distance until the minimum distance become 1. Figure shows one example of the clustering procedure of components.

![Figure 3 Example of Components Clustering](image)

VI. EXPERIMENTS

To examine the validity of our method, DCCC experiments are carried out and the results are contrasted. A dataset of Chinese characters with their components are queried from httpcn.com [9] firstly. The whole count of the dataset is 6881. Then DCCC with our method were performed on a personal computer with Intel i3-2310 CPU and 4 G RAM. The experiment programs are coded with C++ on Visual Studio 2010 platform in Windows 7 system. The process of DCCC expended 1 hour and 12 minutes 37 seconds. All the results of are compared with the exist dataset. About 23 differences were detected. But after checking the differences artificially we found that it’s only some charset difference or blanks in online records. That is to say, our method can decompose Chinese characters exactly at 100% rate. Though the time consume is relative long, but it’s acceptable, as this work is one-off and real-time-need-less.

VII. CONCLUSIONS

In this letter, the problem of DCCC is discussed and a solution for DCCC is proposed. Manual DCCC method is inefficient and error-prone, so DCCC automatically is the only way must be passed. A high accuracy method was proposed in this letter. Experiments result shows that our method is practicable. The speed of our method is not so fast but tolerable. We are going to speed up the algorithm and finish a whole components library of all known Chinese Characters.

The authors acknowledge the support of the National Natural Science Foundation of China (NSFC), Human Motion Features Modeling and Extraction based on Video Flow Model (61262037).

REFERENCES


