

# A Touching Character Database from Tibetan Historical Documents to Evaluate the Segmentation Algorithm

First Author<sup>1</sup>[0000-1111-2222-3333] and Second Author<sup>2</sup>[1111-2222-3333-4444]

<sup>1</sup> Princeton University, Princeton NJ 08544, USA

<sup>2</sup> Springer Heidelberg, Tiergartenstr. 17, 69121 Heidelberg, Germany  
lncs@springer.com

**Abstract.** The benchmarking database plays an essential role in evaluating the performance of touching character string segmentation algorithm. In this paper, we present a new touching Tibetan character string database. Firstly, using the previous proposed layout analysis and text-line segmentation algorithms, we segment scanned images of historical Tibetan documents into text-line images. Then, we find candidate touching character strings using connected component analysis and screen out the correct samples. Finally, we annotate the data manually and establish the touching character database. The database contains 5,844 images two-touching characters and 1,399 images of more than two-touching character. It is applicable to evaluate the segmentation algorithms for touching Tibetan character strings. For each image, the annotated ground truth file includes class labels, candidate segment points, baseline and average stroke width of the Tibetan single character. According to the type of touching, we divide the touching character string into three types: AB, OB and BB. We also count the number of different type of samples and find that 76.27% of the samples belongs to the third type (BB). In the end, we measure the performance of the over-segmentation algorithm on this database for reference.

**Keywords:** Historical Tibetan Documents, Touching Character, Benchmarking Database.

## 1 Introduction

Digitalization of historical documents can protect the literature and improve the reading efficiency. Through an optical character recognition (OCR) system, we can get the content of the literature. A complete OCR system for historical documents includes: image preprocessing, layout analysis, text-line segmentation, character segmentation and character recognition. For the layout analysis of historical Tibetan documents, Zhang et al [1] extract the texts by connected component analysis (CCs) and corner point detection. Li et al [2] propose a baseline-based text-line segmentation algorithm to obtain the text lines of historical Tibetan documents. The research on the segmentation of the touching character string plays an essential role in character segmentation. It is a traditional but not yet fully solved problem, and related researches have started since the 1980s [3]. At present, the segmentation about touching character strings (usually

digital, letters and Chinese character) has achieved satisfactory results, which has important applications in ZIP code recognition, bank check reading and text recognition. In this field, few scholars pay attention to touching Tibetan character string.

Most of the time, researchers use different dataset to verify the segmentation algorithm. Finally, the algorithm proposed by researchers can display good performance in their dataset. It is not accurate to evaluate the performance of different algorithms on different datasets. To compare the efficiency and performance of different algorithms and avoid the impact of different databases, some scholars have established the touching character string benchmarking database. Handwritten touching digital database(HWD-TD) [4] and offline Chinese touching character string database (CASIA-HWDB-T) [5] are the representatives. HWD-TD contains several different kinds of touching and it was generated by connecting 2,000 images of isolated digits extracted from the NIST SD19. However, there is different between factual touching character string and synthesis touching character string. For better evaluate the performance of the segmentation algorithm, Xu et al [5] extracted touching character string from CASIA-HWDB [6] by CCs. CASIA-HWDB-T includes 56,469 touching character strings, most of which are two-touching character, and the 1,818 are multi-touching character.

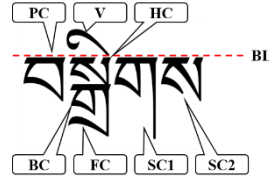
Inspired by the work of Oliveira et al [4] and Xu et al [5], we established a touching Tibetan character string database (TTCS-DB). THCS-TB contains 5,844 images two-touching characters and 1,399 images of more than two-touching character. We have annotated ground truth file for each image, which includes class labels, candidate segment points, baseline and average stroke width of the Tibetan single character. A foreground-based combined algorithm has been carried out on our database. In the following chapter, we will introduce our database in detail.

## 2 Database

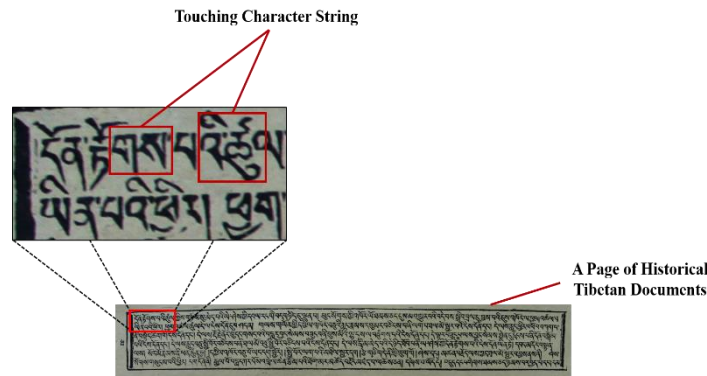
To the best of our knowledge, no database about touching historical Tibetan character string have been built so far. Next, we will introduce the collection and annotation information of the database.

### 2.1 Data collection

A typical Tibetan syllable can be made up of seven consonant letters [7], as shown in **Fig. 1**. There is a base consonant (BC) in each syllable. Other consonants, according to their relative position to the base consonant, are called prefix consonant (PC), head consonant (HC), foot consonant (FC), the first suffix consonant (SC1), the second suffix consonant (SC2) and the vowels (V) respectively. To segment and recognize Tibetan characters, we usually combine the characters in the vertical direction. We scanned the historical Tibetan document named ‘The complete works of Panchen Lama’ and obtained the original scanned images, as shown in **Fig. 2**. We can see that there are many touching character strings in the scanned image.



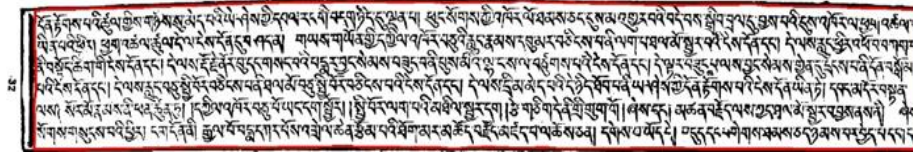
**Fig. 1.** A Typical Tibetan Syllable.



**Fig. 2.** Example of a page of historical Tibetan documents and some touching character string.

In the touching character string extraction, we use the method [1] obtain the text of historical Tibetan documents, and mark the text with a red rectangle, as shown in **Fig. 3 (a)**. Then we divide the text into text-lines by the method [2], as shown in **Fig. 3 (b)**. We can see that different line-texts are distinguished by different colors.

We mark the foreground pixel to 0 and the background pixel to 1. Due to the cause of the ink diffusion and illumination, we deleted the outliers with pixels less than 30 in foreground pixels. At last we collected the original samples through CCs.



(a)



(b)

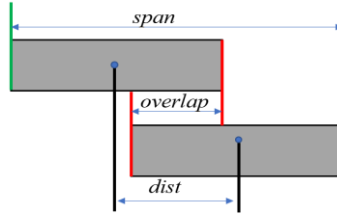
**Fig. 3.** Text-line obtained by method [1] and [2] from historical Tibetan documents.

Considering the overlapping of Tibetan character, we use the algorithm proposed by [8] to combine the characters. The four nearest neighbor pixels were used to mark text-line images, and we save the borders and pixels of each connected area. We can mark the four ends of the boundary as  $x^l$ ,  $x^r$ ,  $y^t$ ,  $y^b$  respectively. We can assume that the boundary information of two connected regions are  $(x_1^l, x_1^r, y_1^t, y_1^b)$  and  $(x_2^l, x_2^r, y_2^t, y_2^b)$ , where  $x_1^l$  less than  $x_2^l$ . According to the formula (1), (2) and (3), we can calculate *ovlp*, *span* and *dist*. *ovlp* represents the length of the overlapping of two components. *span* represents the total length of the two components. *dist* represents the distance of the centroid of the two components. The relationship between *ovlp*, *span* and *dist* can be shown in **Fig. 4**.

$$ovlp = x_1^r - x_2^l \quad (1)$$

$$span = \max(x_1^r, x_2^r) - x_1^l \quad (2)$$

$$dist = \frac{1}{2} |(x_2^l + x_2^r) - (x_1^l + x_1^r)| \quad (3)$$

**Fig. 4.** The relationship between *ovlp*, *span* and *dist*

*nmovlp* is used to measure the degree of overlap, where  $w_1$  and  $w_2$  represent the width of two connected components, respectively.

$$nmovlp = \frac{1}{2} \left( \frac{ovlp}{w_1} + \frac{ovlp}{w_2} \right) - \frac{dist}{span} \quad (4)$$

If  $nmovlp > 0$ , it means that two connected components can be merged. After the whole text-line images processing is completed, the ratio ( $L_r$ ) of the length to width of the average character is calculated. If  $L_r > 1.3$ , it is initially determined to be touching character string. Then, we remove the incorrect samples and obtain the final dataset. **Fig. 5** shows the touching character string extracted from text-line images. In the

following, we will introduce the ground truth file’s format for each touching character string.



**Fig. 5.** Some images extracted from Tibetan history documents, which contains incorrect image. Overlapping characters are marked with red rectangles, the single character is marked by a blue rectangle and error character is marked by in a green rectangle.

## 2.2 Data annotation

All the characters and punctuation in Tibetan language are aligned according to the baseline [2], as shown in **Fig. 1**. This feature is helpful for the segmentation and recognition of Tibetan character. And we divided the touching type into three types, as shown in **Table 1**. The three types are touching points above the baseline (AB), on the baseline (OB) and below the baseline (BB). Through our observation, most of the images in the database are two touching character string. We partition TTCS-DB into two sub-databases according to the number of character in touching character string: TTCS-DB-T and TTCS-DB-M. Each image in TTCS-DB-T contains two characters and TTCS-DB-M is composed of more than two characters, as depicted in **Fig. 6** (a) and (b).

**Table 1.** Touching type of two-touching Tibetan character pair.

Type	Touching Stroke Relation	Examples	Rate (%)
AB			1.37
OB			22.36
BB			76.27



(a)



(b)

**Fig. 6.** Examples of touching character string samples extracted from the database: (a) each image contains two characters from TTCS-DB-T; (b) each image contains more two characters from TTCS-DB-M.

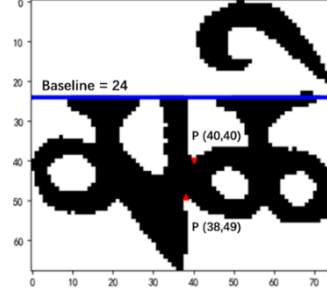
To accurately evaluate the efficiency of the segmentation algorithm, we have annotated the touching character string. The information of the ground truth includes the baseline (BL), the class label (CL), the height and width of the touching character string, the average stroke width (SW), and the candidate segmentation points. BL is an important parameter. The upper vowels are located above the BL, and other characters are under the BL. Using BL to divide the touching characters into two parts can help to improve the accuracy of segmentation. SW and CL are used to evaluate the accuracy of segmentation and recognition respectively. We saved the annotation information in an XML file with the same name as the original image. **Fig. 7** depicts an example of an XML file for a touching character strings.

```

<?xml version='1.0' encoding='utf-8'?>
<root>
  <Page BaseLine="24"
    Class="ཁ, ཆ"
    ImageFileName="bum-18-0836_0_52.png"
    ImageHeight="68"
    ImageStrokeWidth="8"
    ImageWidth="76">
    <TextRegion>
      <Coords>
        <Point x="40" y="40"/>
        <Point x="38" y="49"/>
      </Coords>
    </TextRegion>
  </Page>
</root>

```

(a)



(b)

**Fig. 7.** Example of (a) the annotated information, (b) the touching point (indicated by the red arrow), the baseline (in blue line). The tag TextRegion represents a segmentation path. If the touching character string has two adhesions, TextRegion will have four coordinate points.

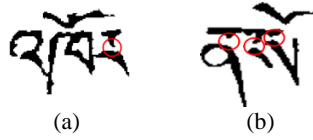
### 2.3 Data analysis

We count the number of characters, touching points and Multi-touching (a segmentation path has multiple points) and total String, as shown in the **Table 2**. In our database, single-touching character string is about ten times than multi-touching character string. For TTCS-DB-M, each touching character string has 2.03 touching points and 3.11 characters on average.

**Table 2.** Statistics of TTCS-DB according to the number of character in touching character string, an overwhelming majority of which is single-touching character string.

Database	#String	#Multi-touching	#Character	#Touch point
TTCS-DB-T	5,844	427	11,688	6,300
TTCS-DB-M	1,399	163	4,350	2,835
Total	7,243	690	16,038	9,135

In follow-up investigation, we found a common phenomenon. Due to the degradation of historical Tibetan documents, the strokes of character are broken, as shown in the **Fig. 8**. When we annotate data, we spend a lot of time to identify touching character string. In the character recognition for Tibetan, broken strokes will bring great challenge.



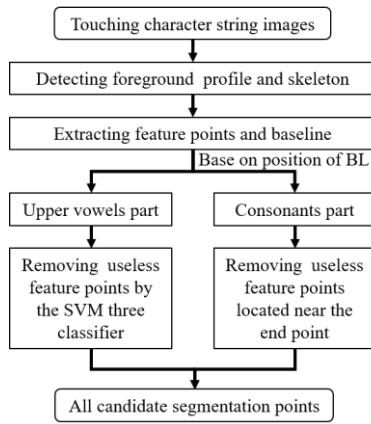
**Fig. 8.** Example of the broken strokes in the touching character string (in the red ring).

### 3 Algorithm

The segmentation algorithm can be roughly divided into two categories, implicit segmentation algorithm and explicit segmentation algorithm [9]. The main idea of implicit segmentation algorithm is to traverse the touching character string from left to right to get a feature sequence by a narrow sliding window. Then based on the HMM of the statement, the character recognition and segmentation results of the entire text line are obtained. The explicit segmentation algorithm divides the touching character string into multiple substrings according the feature points in the image. It can be further divided into two categories, one is weak-segmentation and the other is over-segmentation. The main feature of the weak segmentation algorithm is that only one segmentation path is generated, which is suitable for less touching. The representative algorithm includes vertical projection [10], drip algorithm [11] water reservoir [12] and so on. The over-segmentation algorithm produces multiple segmentation paths. It can be roughly divided into three categories: foreground-based [13] [14], background-based [15], and recognition-based [16].

We have measured the performance of an over-segmented algorithm on this database for reference. The over-segmented algorithm is based on feature point detection.

The flowchart of our algorithm is shown in **Fig. 9**. Firstly, the foreground profile and skeleton are detected. Secondly, we detect the feature points on the profile by algorithm [17] and detect the baseline of touching character string. According to the Tibetan baseline, we divide the touching character string into two parts: upper vowels and consonants. In the end, we will remove all the useless feature points. For the upper vowels part, we use feature points directly to segment upper vowels. Then, we designed a support vector machine (SVM) classifier [18] to predict the probability that the image is a vowel. When the probability of each parts is acceptable, we keep this feature point, otherwise we delete it. For the consonant part, all the feature points located near the end point are deleted according to the skeleton information of the touching character string.

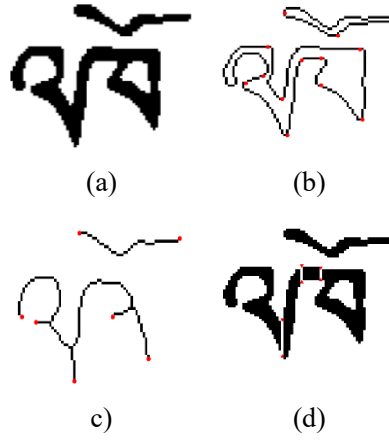


**Fig. 9.** The flowchart of our segmentation algorithm.



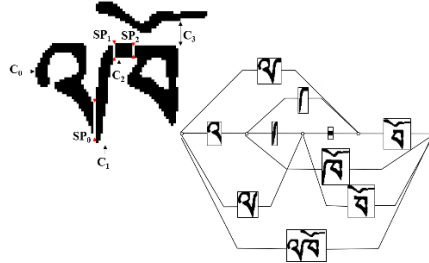
## 4 Experiments

We extract the connected components by 8-connected regions for each image, and we delete components where width and height less than  $SW*2$ . **Fig. 10** shows the details and candidate segment points by our algorithm. We can see that our feature points do not appear in pairs. We have chosen two ways to build segmentation paths. When two feature points are located on either side of the stroke, we connect the two feature points to form a segmentation path. In other cases, we cut the strokes directly based on the feature point to form a segmentation path.



**Fig. 10.** Example of (a) original touching character string, (b) foreground profile and feature points, (c) foreground skeleton and end points in skeleton, (d) segmentation path. We remove useless points by SVM classifier and end points in skeleton.

**Fig. 11** shows an example of an image segmented by our algorithm and its corresponding segmentation graph. Three paths ( $SP_0$ ,  $SP_1$  and  $SP_2$ ) and four components ( $C_0$ ,  $C_1$ ,  $C_2$  and  $C_3$ ) be generated in the end. According to Tibetan character characteristics, we assume that a Tibetan character can be composed of three components at most. The touching character string can produce ten sub-images. We need use the classifier to score ten sub-images and find the largest score path in the graph to represent the final segmentation and recognition results.



**Fig. 11.** Example of a segmentation graph.

We evaluate the performance of the algorithm based on the distance ( $d$ ) between a touching point and a candidate point. When  $d$  is less than a threshold  $d_{th}$ , we think that the candidate point is a correct segmentation point. In our paper, we set  $d_{th}$  equal to  $1.4 \times SW$ . We also calculate the recall rate  $R$  and the precision rate  $P$  [4] to evaluate our algorithm, as following.

$$R = \frac{\text{\#the number of correct separating points}}{\text{\#the number of total truth touching points}} \times 100\% \quad (5)$$

$$P = \frac{\text{\#the number of correct separating points}}{\text{\#the number of total candiate spatating points}} \times 100\% \quad (6)$$

**Table 3** reports the performance of the foreground-based combined segmentation algorithm on the proposed database. In our algorithm, we extracted the Tibetan baseline with an accuracy rate of 95%. Since we forcibly split upper vowels and consonants, the actual segmentation result is better than the calculated value. Over-segmentation algorithm can achieve better segmentation results, but too many candidate points will bring expensive calculations. **Table 4** reports the average number of candidate points generated by our algorithm and the time to process each file in Python program.

**Table 3.** Performance of the foreground-based combined algorithm on the database.

Database	R (%)	P (%)
TTCS-DB-T	87.54	27.56
TTCS-DB-M	80.78	32.98
Average	86.60	30.25

**Table 4.** The average number of candidate points and time to process each file.

Database	Average number	Time for each file (s)
TTCS-DB-T	3.22	0.0905
TTCS-DB-M	5.21	0.1286
Average	3.60	0.0978

## 5 Conclusion and Future Works

In this paper, we introduce the database about how to obtain the touching Tibetan character string from historical Tibetan documents and the ground truth file's format for each touching character string in detail. The database we have established can be used to evaluate segmentation algorithms for touching Tibetan character string. We have implemented a foreground-based combined segmentation algorithm in the database and analyzed the experimental results. 86.60% of the samples can be correctly segmented and a touching character string generates 3.6 candidate points on average. In the future, we hope to improve the precision of the algorithm and extend our database further. Meanwhile, we will evaluate other segmentation algorithms in our database for reference.

## References

1. Zhang X, Duan L, Ma L, et al.: Text Extraction for Historical Tibetan Document Images Based on Connected Component Analysis and Corner Point Detection. In: 2nd Chinese Conference on Computer Vision, CCCV 2017, pp. 545-555. Springer Verlag, Tianjin, China (2017).
2. Li Y, Ma L, Duan L, et al.: A Text-Line Segmentation Method for Historical Tibetan Document Based on Baseline Detection. In: 2nd Chinese Conference on Computer Vision, CCCV 2017, pp. 356-367. Springer Verlag, Tianjin, China (2017).
3. Casey R G, Lecolinet E.: Survey of Methods and Strategies in Character Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7), 690-706 (1996).
4. Oliveira L S, Britto A S, Sabourin R.: A Synthetic Database to Assess Segmentation Algorithms. In: 8th International Conference on Document Analysis and Recognition, pp. 207-211. Inst. of Elec. and Elec. Eng. Computer Society, 445 Hoes Lane - P.O.Box 1331, Piscataway, NJ 08855-1331, United States, Seoul, Korea, Republic of (2005).
5. Xu L, Yin F, Wang Q F, et al.: A Touching Character Database from Chinese Handwriting for Assessing Segmentation Algorithms. In: 13th International Conference on Frontiers in Handwriting Recognition, ICFHR 2012, pp 89-94. IEEE Computer Society, 10662 Los Vaqueros Circle - P.O. Box 3014, Los Alamitos, CA 90720-1314, United States, Bari, Italy (2012).
6. Liu C L, Yin F, Wang D H, et al.: CASIA Online and Offline Chinese Handwriting Databases. In: 11th International Conference on Document Analysis and Recognition, ICDAR 2011, pp. 37-41. IEEE Computer Society, 445 Hoes Lane - P.O.Box 1331, Piscataway, NJ 08855-1331, United States, Beijing, China (2011).
7. Huang, H., Da, F.: General structure based collation of Tibetan syllables. *J. Inf. Comput.* 6(5), 1693-1703 (2010)
8. Liu C L, Koga M, Fujisawa H.: Lexicon-driven handwritten character string recognition for Japanese address reading. In: 6th International Conference on Document Analysis and Recognition, ICDAR 2001, pp. 877-881. IEEE Computer Society, Seattle, WA, United states (2001).
9. Rehman A, Mohamad D, Sulong G. Implicit vs explicit based script segmentation and recognition: a performance comparison on benchmark database. *International Journal of Open Problems in Computer Science & Mathematics*, (3) 352-364 2009.
10. Chitrakala S, Mandipati S, Raj S P, et al.: An Efficient Character Segmentation Based on VNP Algorithm. *Research Journal of Applied Sciences, Engineering and Technology*, 4(24) 5438-5442 (2012).
11. Congedo G, Dimauro G, Impedovo S, et al.: Segmentation of numeric strings. In: Proceedings of the Third International Conference on. IEEE Computer Society, 1028-1033 (1995).
12. Pal, U., Belaid, A., & Choisy, C.: Touching numeral segmentation using water reservoir concept. *Pattern Recognition Letters*, 24(1), 261-272 (2003).
13. Jayarathna U K S, Bandara G E M D C.: A Junction Based Segmentation Algorithm for Offline Handwritten Connected Character Segmentation. In: CIMCA 2006: International Conference on Computational Intelligence for Modelling, Control and Automation, Jointly with IAWTIC 2006: International Conference on Intelligent Agents Web Technologies and International Commerce. pp. Inst. of Elec. and Elec. Eng. Computer Society, 445 Hoes Lane - P.O.Box 1331, Piscataway, NJ 08855-1331, United States, Sydney, NSW, Australia (2006).
14. Xu L, Yin F, Liu C L.: Touching Character Splitting of Chinese Handwriting Using Contour Analysis and DTW. In: 2010 Chinese Conference on Pattern Recognition, CCPR, pp 814-

818. IEEE Computer Society, 445 Hoes Lane - P.O.Box 1331, Piscataway, NJ 08855-1331, United States, Chongqing, China (2010).
15. Lu Z, Chi Z, Siu W, et al.: A background-thinning-based approach for separating and recognizing connected handwritten digit strings. *Pattern Recognition*, 32(6): 921-933 (1999).
16. Cheung A, Bennamoun M, Bergmann N W.: An Arabic optical character recognition system using recognition-based segmentation. *Pattern Recognition*, 34(2):215-233 (2001).
17. Shi J, Tomasi.: Good features to track. In: *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 593-600. Publ by IEEE, Los Alamitos, CA, United States, Seattle, WA, USA (1994).
18. Chen J, Takagi N.: Gray-Scale Morphology Based Image Segmentation and Character Extraction Using SVM. In: *46th IEEE International Symposium on Multiple-Valued Logic, ISMVL 2016*, pp.177-182. IEEE Computer Society, Sapporo, Hokkaido, Japan (2016).