

Local Segmentation of Touching Characters using Contour based Shape Decomposition

Le Kang, David Doermann
 Institute for Advanced Computer Studies
 University of Maryland
 College Park, MD, USA
 {lekang, doermann}@umiacs.umd.edu

Huaigu Cao, Rohit Prasad, Prem Natarajan
 Raytheon BBN Technologies
 Cambridge, MA, USA
 {hcao, rprasad, pnataraj}@bbn.com

Abstract—We propose a contour based shape decomposition approach that provides local segmentation of touching characters. The shape contour is linearized into edgelets and edgelets are merged into boundary fragments. The connection cost between boundary fragments is obtained by considering local smoothness, connection length and a stroke-level property called the Same Stroke Rate. Samples of connections among boundary fragments are randomly generated and the one with the minimum global cost is selected to produce the final segmentation of the shape. To obtain a bipartite segmentation using this approach, we perform an iterative search for the parameters that finally yields two components on a shape. Experimental results on synthetic shape images and the LTP dataset show that this contour based shape decomposition technique is promising and it is effective for providing local segmentation of touching characters.

Keywords—touching; segmentation; shape decomposition; smoothness

I. INTRODUCTION

Robust segmentation of touching characters remains an open problem in offline handwritten document analysis, despite extensive research. Generally an integrated segmentation-evaluation strategy is employed where multiple segmentation hypotheses are first generated by analysis of contour, skeleton, background or projection profile, then an optimal decision is selected by recognition or other evaluation methods.



Figure 1. Challenging touching samples

We are interested in the generation of local segmentation hypotheses. In the literature, skeleton and foreground/background analysis methods [1-3] which detect intersections on skeletons, are prone to error caused by inadequate thinning. Contour analysis methods [4-6] use angle and curvature features, and propose multiple hypotheses. A set of rules that consider global configurations have to be designed, and they often depend on the

application. Using only simple local properties of contour, it is difficult to process complicated touching cases without context like those shown in Fig 1.

Consider however how humans make judgments for this problem. Our eyes trace the contour and decompose the touching shape into smooth strokes with maximum homogeneity. So we believe a reasonable segmentation on a local touching region is often possible, if we can fully exploit contour and stroke properties.

In this paper we propose a contour based shape decomposition method that can be used to make local segmentation decisions on touching characters. Inspired by the contour analysis in [8], the decomposition starts with a piecewise linear representation of “edgelets”. Edgelets are then grouped into boundary fragments that are basic units used to generate the segmentation. To compute connection costs between each pair of boundary fragments, we consider local smoothness, connection length, and a stroke level property called the Same Stroke Rate. Samples of connections among boundary fragments are randomly generated and the one with the minimum global cost is selected to produce the final segmentation of the shape. This shape decomposition method is able to segment a character shape into multiple strokes. To apply this method to obtain a binary segmentation on a local touching pattern, we perform an iterative search for the parameters that yield exactly two result components.

II. SHAPE DECOMPOSITION

A. Edgelets and boundary fragments

To generate the edgelet representation we employ the Ramer-Douglas-Peucker algorithm [11] to fit contour points to a simplified polygon. The only parameter of the algorithm is the tolerance that represents the largest distance a point deviates from its fitted line, and is chosen as 1/5 stroke width of the connected component, which is estimated by a distance transform. We refer to each edge on the polygon as an edgelet and define the turning angle as the angle between neighboring edgelets. Edgelets then are grouped into boundary fragments considering the saliency of turns, i.e. boundary fragments only break at salient turns. Assume two edgelets are denoted as AB and BC , and the turning angle between them is φ , then the turn is salient if and only if

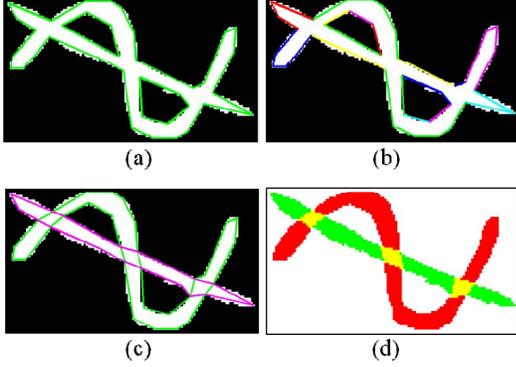


Figure 2. (a)Edgelets. (b)Boundary fragments in different colors. (c) Polygons. (d) Resulting components (yellow for overlapping).

$$\varphi > T_1, \text{ or } \varphi \times \min(|AB|, |BC|) > T_1 \times SW, \quad (1)$$

where T_1 is a threshold and SW denotes the stroke width. Empirically we set $T_1 = \pi/6$. The intuition here is that either a large turning angle or a small angle with two large edgelets will indicate a salient turn. One more constraint is that the boundary fragment will not break at a turn towards the inside of the shape. Fig 2(a) and (b) show an example of edgelets and boundary fragments obtained for a shape contour.

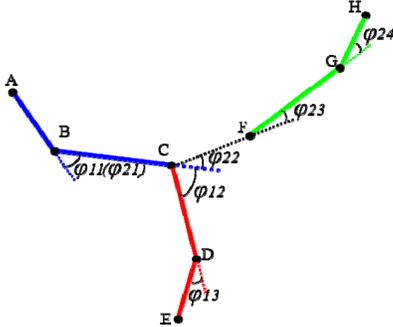


Figure 3. Example of turning angles and immediate/across connection.

B. Connections between boundary fragments

Each boundary fragment has two ends, and each end is to be connected to another. There are two kinds of connections between ends— immediate and cross. An immediate connection occurs when two neighboring boundary fragments share the same end point. A cross connection occurs when two ends are separate and connected by a virtual edgelet. Consider the example in Fig 3, ABC , CDE and FGH are three boundary fragments. If ABC and CDE are connected at C , then the two boundary fragments are connected by an immediate connection. If ABC and FGH are connected by the virtual edgelet CF , it is a cross connection.

Spline interpolation is used in [8] to generate virtual connections, and the cost of a virtual connection c_{virt} is measured by the integral of curvature square as follows

$$c_{virt} = \int (d\theta/ds)^2 ds, \quad (2)$$

where ds and $d\theta$ are infinitesimals of curve length and angle respectively. However in practice we find such a measure is not scale-invariant. For two curves with similar shape but different sizes, the larger and longer one has smaller cost under that measure.

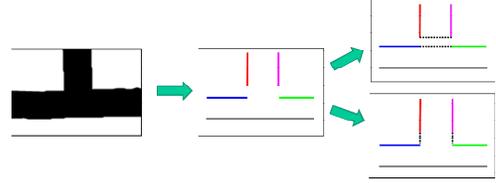


Figure 4. Two configurations of connections among boundary fragments.

In our approach, we measure the connection cost c_{cost} as a product of four terms that describe both local and global relations between boundary fragments:

$$c_{cost} = c_{ang} \times c_{var} \times c_{len} \times c_{SSR}. \quad (3)$$

We will explain each term in the following paragraphs.

The first term c_{ang} is a concave function of the total turning angle at the connection. For an immediate connection the total turning angle is simply the turning angle at the connection point, while for a cross connection the total turning angle is interpreted as the sum of the two turning angles formed by the virtual edgelet connecting the two boundary fragment ends. In Fig 3, φ_{12} is the total turning angle for the immediate connection at C , and $(\varphi_{22} + \varphi_{23})$ for the cross connection CF . If the total turning angle is φ , the then its angle term $c_{ang}(\varphi)$ is expressed as

$$c_{ang}(\varphi) = \sqrt{\varphi/(2\pi) + s}, \quad (4)$$

where s is a small positive offset preventing zero cost. We choose to use a positive concave function here. Fig 4 shows an example of two possible configurations of connections for four boundary fragments. If we only consider the angle term, the upper configuration leads to a total cost of $c_1 = c_{ang}(0) + c_{ang}(\pi)$, and the lower one of $c_2 = c_{ang}(\pi/2) + c_{ang}(\pi/2)$. The upper configuration is preferred for shape segmentation, i.e. we need $c_1 > c_2$, thus a concave function for the angle term is required.

The total turning angle is an intuitive measure that reflects the local smoothness where a connection is made. But in reality when the contour of a shape is noisy, the total turning angle may be large even if two boundary fragments are supposed to have a globally smooth connection. Thus a non-local measure of contour smoothness is needed.

The second term is the angular variance c_{var} . In [10] curve smoothness is measured by the angular variance to select the optimal solution in segmenting touching strokes between text lines. For our problem the angular variance term involves either one or two turning angles at the

connection, and two supporting angles. The two supporting angles are the angles next to turning angles. Consider the example in Figure 3, φ_{11} and φ_{13} are the supporting angles for the immediate connection at C , while φ_{21} and φ_{24} are the supporting angles for the cross connection CF . The angular variance term c_{var} is calculated as the standard deviation of the three or four angles. This term considers the context of the connection and reflects the long range smoothness.

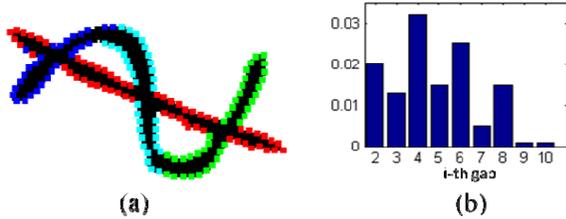


Figure 5. (a)Ncuts clustering with 4 clusters on contour sample points. (b) Gaps of eigenvalues. The i th gap means the difference between i th and $(i+1)$ th eigenvalue. The largest gap omitted in the figure.

The third term c_{len} is a penalty on the length of the connection. The length is the distance between two ends of a cross connection, normalized by the maximum stroke width. The length is 0 for an immediate connection. An exponential function of the length is used to compute c_{len} as

$$c_{len}(L) = \beta + \alpha \times \exp(-\gamma L), \quad (5)$$

where β is an offset, α is a scaling factor and L is the connection length. Empirically we set $\alpha = 0.1$, $\beta = 1$. The term penalizes on larger lengths but is not sensitive to small normalized lengths that are less than 1. This design allows short cross connections, while preventing connections that are too long.

The fourth term c_{SSR} is a function of the Same Stroke Rate (SSR). SSR measures how likely it is that two edgelets belong to the same stroke. To compute SSR, we first sample points on the shape contour with a sample interval distance equal to 1/5 of stroke width. Then we set up a similarity matrix for all the points. If two points are visible to each other inside the shape, their similarity is 1, otherwise the similarity is set to a small positive value, e.g. 2^{-8} . Then the similarity matrix is fed to Normalized Cuts (Ncuts) [12]. There is no way to get the generally optimal cluster number without assumptions of underlying model, but [13] suggests choosing the number k such that all eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_k$ are very small but λ_{k+1} is relatively large. In practice we first choose an initial guess $k=10$ (larger than possible number of clusters), determine the k according to the second largest gap between the eigenvalues, and rerun the Ncuts with k . Figure 5 (a) and (b) illustrate the clustering result on contour sample points of a shape and the corresponding eigenvalue gaps excluding the largest one. Let u and v denote the sets of sample points on two edgelets at the ends of boundary fragments respectively and S_i stands for cluster i , then the SSR of u and v is computed as follows:

$$p_u(i) = |\{x: x \in u, x \in S_i\}| / |\{x: x \in u\}|, \quad (6)$$

$$SSR(u, v) = \max_i p_u(i) p_v(i), \quad (7)$$

where $|\cdot|$ denotes the cardinality of a set. A larger SSR indicates that more sample points in u and v belong to same cluster and the two edgelets are more likely to be connected. Let $c_{SSR}(u, v) = SSR(u, v)^{-1}$. Thus c_{SSR} provides another non-local measure for connection smoothness.

C. Sampling Connections

To reduce the computational load and prevent invalid connections, two types of connections are not allowed. One is the connections between boundary ends that are not visible to each other inside the shape. The other is connections that cause a topological conflict such as merging background and foreground.

We define the affinity of a connection as the exponential function of the negation of cost

$$affinity = \exp(-\gamma \times c_{cost}), \quad (8)$$

where γ is a scaling factor. For each end, the affinities are normalized to a unit sum so that a distribution is obtained to describe the probability of connecting to every other end.

We randomly select an open boundary fragment end, connect it with another open end according to its affinity distribution. Note that conditional distributions of affinity are used at each end, i.e. only open ends are considered. This procedure is repeated until no end is open. Each end is connected to only one other. During the procedure we check each newly resulting polygon once it is formed and reject this sample if that polygon is self-intersecting.

Given N boundary fragments we take N^2 samples as suggested by [8]. Among all the accepted samples, we choose the one with the least global cost as the final configuration. It is possible that all samples are rejected, then we only make immediate connections and one single polygon is obtained.

D. Pixel assignment

The global configuration only gives us polygons that reflect approximately how the shape contour is divided. Pixels within or on the boundaries of different polygons obviously belong to different sub components of the shape. To complete the real segmentation on the shape, we need to assign pixels that are outside every polygon to one of the polygons. For each polygon, we apply the distance transform to find the distance from each outside pixel to the polygon. Then for each outside pixel we assign it to the polygon with the shortest distance. Eventually we segment the shape into sub components that are not mutually exclusive but collectively exhaustive. Figure 7 shows an example of sub components obtained from polygons.

III. BIPARTITE SEGMENTATION

After the shape decomposition, we may obtain either a single component or multiple components. To enforce a

bipartite segmentation, we need to limit the number of resulting components to two.

A practical method is to alter the threshold T_1 used in merging edgelets into boundary fragments. We use an iterative search to find the parameters that yield only two components, as shown in Fig 6.

```

N : number of resulted components
Seg( $T_1$ ) : do segmentation with parameter  $T_1$ 
L : left bound of  $T_1$ 
R : right bound of  $T_1$ 
M: maximum iteration allowed

WHILE iteration number less than M
  N = Seg( $T_1$ );
  IF N > 2
    L =  $T_1$ ;
    IF R is not set
       $T_1$  =  $T_1 * 2$ ;
    ELSE
       $T_1$  = ( $T_1$  + R)/2;
  ELSEIF N < 2
    R =  $T_1$ ;
    IF L is not set
       $T_1$  =  $T_1 / 2$ ;
    ELSE
       $T_1$  = ( $T_1$  + L)/2;
  ELSE
    break;

```

Figure 6. Iterative search for the parameter T_1 .

Experimentally, we observed that complicated touching samples usually need more iterations to achieve a binary segmentation and get lower accuracy. Based on this fact we consider the iteration number as a reflection of confidence for the segmentation. A larger iteration number indicates lower confidence in the segmentation. With this measure of confidence we can efficiently generate segmentations only with high confidence by setting a threshold for iteration number, and leave difficult samples for other segmentation techniques that are finer and cost more time.

IV. EXPERIMENTAL RESULTS

We first apply our shape decomposition to a number of synthetic shape images. A few examples are shown in Fig 7. As we can see the proposed approach is able to decompose a shape into stroke-level components.

We applied the proposed approach with binary segmentation adaption on the LTP dataset [14]. There are 744 local touching patches each for training and testing respectively. The evaluation protocol is the same as [14] using MatchScore, and segmentation with a MatchScore above 0.80 is considered as a correct one. We optimized the parameters on the train data and evaluated the performance on the testing data.

Performances under different rejection thresholds were recorded. As illustrated in Fig 8, rejecting all samples with more than one iterations yields the maximum accuracy on accepted samples, and the accuracy drops as the rejection

threshold increases. This result demonstrates that the rejection strategy is effective.

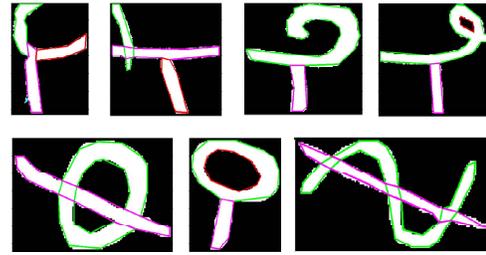


Figure 7. Examples of synthetic shapes segmented into stroke-level components.

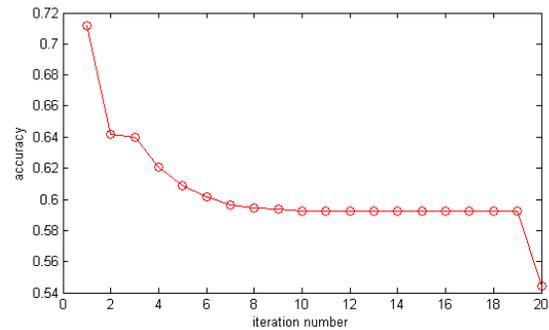


Figure 8. The accuracy of accepted segmentation varies with rejection

In Table I we compare the accuracy and efficiency of the proposed method with the template based method [14]. We can see the proposed method consumes significantly less time than [14] and its performance is fairly good when rejecting about half of the samples.

TABLE I. ACCURACY AND EFFICIENCY

	<i>Template based</i> [14]	<i>Proposed</i>	
		Rejection rate/ threshold	Accuracy
Rejection rate/ threshold	0%	0%	57.1% / 1 iteration
Accuracy	71.4%	54.7%	71.2%
Relative average time cost	1	0.051	0.014

Fig 9 shows a few segmentation examples. Some difficult examples are segmented fairly well, with resulting components being smooth strokes.

V. CONCLUSION

We proposed a contour based shape decomposition approach that provides local segmentation of touching characters. On synthetic data, the proposed method decomposed shapes into stroke-level components. Experiments carried on LTP dataset showed that the proposed approach is efficient and by reasonably rejecting part of samples, it achieves a fairly good accuracy on providing local segmentation of touching characters.



Figure 9. Segmentation examples. Yellow indicates the part is shared by red and green.

ACKNOWLEDGMENT

The partial support of this research by DARPA through BBN/DARPA Award HR0011-08-C-0004 under subcontract 9500009235, the US Government through NSF Award IIS-0812111 is gratefully acknowledged.

REFERENCES

- [1] Y.K. Chen, and J.F. Wang. "Segmentation of single- or multiple-touching handwritten numeral string using background and foreground analysis," IEEE TPAMI, 22(11):1304-1317, 2000.
- [2] H. Ikeda, Y. Ogawa, M. Koga, H. Nishimura, H. Sako, and H. Fujisawa. "A recognition method for touching Japanese handwritten characters," ICDAR'99, pp. 641-644, 1999.
- [3] S. Wshah, Z. Shi, and V. Govindaraju, "Segmentation of Arabic handwriting based on both contour and skeleton segmentation," ICDAR'09, pp. 793-797, 2009.
- [4] Z. Shi, S.N. Shrihari, Y.-C. Shin, and V. Ramanaprasad, "A System for Segmentation and Recognition of Totally Unconstrained Handwritten Numeral Strings," ICDAR'97, vol. 2, pp. 455-458, 1997.
- [5] N.W. Strathy, C.Y. Suen, and A. Krzyza, "Segmentation of Handwritten Digits Using Contour Features," ICDAR'93, pp. 577-580, 1993.
- [6] C.-L. Liu, M. Koga, H. Fujisawa. "Lexicon-driven segmentation and recognition of handwritten character strings for Japanese address reading," IEEE TPAMI, 24(11): 1425-1437, 2002.
- [7] D. Yu and H. Yan, "Separation of touching handwritten multi-numeral strings based on morphological structural features," Pattern Recognition, 34(3): 587-599, 2001.
- [8] C. Liu, W. T. Freeman and E. H. Adelson, "Analysis of contour motions," NIPS'06, pp. 913-920, 2006.
- [9] X. Ren, C. Fowlkes, and J. Malik, "Learning Probabilistic Models for Contour Completion in Natural Images," IJCV, 77(1-3): 47-63, 2008.
- [10] N. Ouwayed and A. Belaïd, "Separation of Overlapping and Touching Lines within Handwritten Arabic Documents," CAIP'09, pp. 237-244, 2009.
- [11] Douglas, D., Peucker, T., "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," The Canadian Cartographer, 10(2): 112-122, 1973.
- [12] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," PAMI, 22(8): 888-905, 2000.
- [13] U. von Luxburg, "A Tutorial on Spectral Clustering," Statistics and Computing 17(4): 395-416, 2007.
- [14] Le Kang and David Doermann. "Template based Segmentation of Touching Components in Handwritten Text Lines." ICDAR'11, pp. 569-573, 2011.