15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

A Sampling Method for Processing Contours Drawn with an Uncertain Stroke Order and Number

Kazuya Ose, Kazunori Iwata, and Nobuo Suematsu
Hiroshima City University
3-4-1 Ozuka-Higashi, Asa-Minami-Ku, Hiroshima, 731-3194 Japan
mb67004@e.hiroshima-cu.ac.jp, {kiwata,suematsu}@hiroshima-cu.ac.jp

Abstract

Although there are several effective methods for placing sample points on contours drawn with a certain stroke order and number, little attention has been paid to methods for placing sample points on contours with an uncertain stroke order and number. In this paper, we place sample points appropriately on such contours using an optimization method. The underlying idea is to take into account the degree in graph theory and edge length in geometry for the optimization. Using a dataset of line drawings, we confirm that our method provides an appropriate placement of sample points in terms of shape retrieval.

1 Introduction

Consider extracting some points from the contours of a shape that are described by line drawings using a pen or are boundaries obtained through image segmentation. The extracted points are called sample points. Typically, instead of contours, we use sample points to reduce the computational cost of processing the shape. The contours are represented as a set of points. The contours have a certain stroke order and number when they represent letters, such as the English alphabet and Arabic figures. However, generally, they have an uncertain order and number, for example, when they represent a picture of a symbol. Although vertex detection using a polygonal line approximation (PLA) for curves, such as [1, 2], and locating high curvature points, which were introduced in [3, 4, 5], should be effective for contours drawn with a certain stroke order and number, they do not work well for contours drawn with an uncertain stroke order and number because of their dependence on the stroke order and number. Little attention has been paid to placing sample points on such contours. Thus, the purpose of this paper is to provide an optimization method, which is a type of direct search [6], for placing sample points appropriately on such contours. This method will allow for the analysis of a greater variety of shapes. Using a dataset of line drawings, we demonstrate that our method yields a good placement of sample points in terms of shape retrieval.

This paper is organized as follows: We start by considering where sample points should be located on such contours and then present our method in Section 2. Using a dataset of line drawings, we examine our method in Section 3. Finally, we provide a summary in Section 4.

2 Placement of Sample Points

The first problem to be clarified is where sample points should be located on the contours of line drawings drawn with an uncertain stroke order and number. As a similar problem, we can extract characteristic points from contours obtained by a thinning operation [7] for digital image curves. In that extraction, all points (pixels) that consist of a contour are divided to the end, branch, or passing points, and then the contours are represented by segments based on the end and branch points. This implies that the end and branch points are characteristic of the contours. In this paper, we further develop the characteristic points to make sample point-based shape analysis easier. Specifically, we regard sample points as vertices of a graph over the contours. Then, each vertex should be located on a point that has a high degree (the number of edges incident to it [8]) and long edge. Such a point plays a vital role in representing the shape of contours. Thus, the aim of this paper is to automatically place sample points at such locations for contours drawn with an uncertain stroke order and number.

2.1 Notation

Let $\mathcal{S}$ be a set of points that consist of contours of a shape. We assume that $\mathcal{S}$ is a finite set. For example, when we consider a set of pixels of digital image contours, it satisfies this assumption. When we consider contours in the x-y plane, extracting finite points from the contours provides a finite set. We denote n sample points by $(x_1, y_1), \ldots, (x_n, y_n)$, and assume that $3 \leq n \leq |\mathcal{S}|$ holds.

In fact, the placement problem for n sample points falls into a combinational optimization problem when we choose n sample points from the set of points. Clearly, the total number of choices is $\binom{|\mathcal{S}|}{n}$. Depending on $|\mathcal{S}|$ and n, that number can be very large; thus, we cannot process all the choices. Hence, in this paper, we present a search method for efficiently determining a good choice that corresponds to a placement. The search method is an optimization without derivative information, which makes it suitable for problems with non-smooth objective functions. Additionally, it is quite simple and easy to use.

2.2 Method

Consider a complete graph K_n given by n vertices [8]. The n vertices are fixed by the coordinates of n sample points, that is, $(x_1, y_1), \ldots, (x_n, y_n)$ in $\mathcal{S}$. For simplicity, let $\mathcal{P}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. All the edges of K_n have a common width (thickness) denoted by w, and the set of points on all the edges

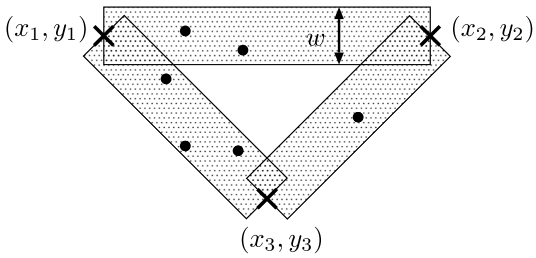

Figure 1. Set of points on all edges with a common width w.

is given by $\mathcal{E}\,(w\,;\,\mathcal{P}_n)$. Figure 1 illustrates an example of $\mathcal{E}\,(w\,;\,\mathcal{P}_3)$. In this figure, the crosses represent vertices and dots represent points in a set $\mathcal{S}$. The shaded portions represent edges with a common width w. Then, $|\,\mathcal{E}\,(w\,;\,\mathcal{P}_3)\,| = 6$ in this example. For any $i \in \{1, \ldots, n\}$, deleting the i-th vertex at (x_i, y_i) yields a complete graph K_{n-1} whose $n-1$ vertices are given by $\mathcal{P}_n \setminus \{(x_i, y_i)\}$, where $\setminus$ denotes the set difference. For simplicity, let $\mathcal{P}_{n-1}^{(x_i,y_i)} = \mathcal{P}_n \setminus \{(x_i, y_i)\}$ again. Then, $\mathcal{E}\left(w\,;\,\mathcal{P}_{n-1}^{(x_i,y_i)}\right)$ denotes the set of points on all the edges of K_{n-1}. For example, regarding the bottom sample point in Figure 1, we obtain $\left|\,\mathcal{E}\left(w\,;\,\mathcal{P}_2^{(x_3,y_3)}\right)\right| = 2$. We then evaluate the contribution to the i-th vertex using

$$f\,(x_i, y_i\,;\,w, \mathcal{P}_n) = \left|\,\mathcal{E}\,(w\,;\,\mathcal{P}_n) \setminus \mathcal{E}\left(w\,;\,\mathcal{P}_{n-1}^{(x_i,y_i)}\right)\right|. \tag{1}$$

Using this contribution, we perform a direct search method to locate n vertices. Specifically, given the set of points $\mathcal{S}$:

1. Fix n and w, which denote the number of vertices and edge width, respectively.

2. Select n initial vertices, $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{S}$, which are different.

3. Repeat the following steps while $s < s'$ holds.

   (a) Memorize the coordinates of the current vertices, $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{S}$, and construct $\mathcal{P}_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

   (b) Set s to be the sum of contributions $\sum_{i=1}^{n} f\,(x_i, y_i\,;\,w, \mathcal{P}_n)$.

   (c) For all $i \in \{1, \ldots, n\}$, based on $\mathcal{P}_{n-1}^{(x_i,y_i)}$, replace (x_i, y_i) in $\mathcal{P}_n$ with

   $$(x', y') = \underset{(x,y) \in \mathcal{S} \setminus \mathcal{P}_{n-1}^{(x_i,y_i)}}{\operatorname{argmax}} f\,(x, y\,;\,w, \mathcal{Q}_n(x, y))\,, \tag{2}$$

   where $\mathcal{Q}_n(x, y) = \mathcal{P}_{n-1}^{(x_i,y_i)} \cup \{(x, y)\}$.

   (d) Set s' to be the sum of contributions using the replaced elements in $\mathcal{P}_n$.

4. Draw the memorized vertices as sample points.

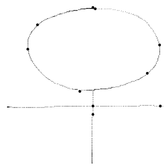

Figure 3. Example of the resultant placement using the PLA: a feminine-shaped line drawing.

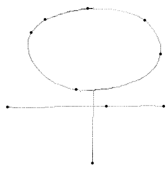

Figure 4. Example of the resultant placement using our method: a feminine-shaped line drawing.

Thus, based on the memorized vertices, we place sample points on the contours. Note that the sum of contributions corresponds to the non-smooth objective function in this optimization problem. The objective function is designed to place a sample point on the location that has a high degree and long edge; thus, it is an effective location for representing the shape of contours.

3 Experiments

In this section, we compare the placement of sample points provided by a typical PLA and our method.

3.1 Dataset

Using software available from [9], we created a dataset that consisted of 50 line drawings, each of which represented a shape and was classified into one of five classes. There were 10 shapes in each class. Figure 2 illustrates all the line drawings. The dataset can be obtained from [10]. The line drawings in each class were drawn using a different stroke order, stroke number, and shape scale. Moreover, they were expressed by different numbers of points. These settings allow for an examination of the effect of the difference in stroke order and number on our method.

We used a mixture of von Mises distributions (vMDs) [11] as a local descriptor of shape because this mixture is invariant with respect to stroke order, stroke number, and shape scale. The details regarding the matching cost with vMDs may be found in [11]. The concentration parameter κ for determining the vMDs was fully tuned for the PLA and our method; thus, it was set to $\kappa = 13$ in the PLA and $\kappa = 18$ in our method. For all the line drawings, the number of sample points was fixed to $n = 10$.

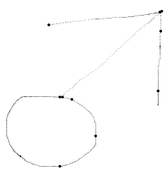

Figure 5. Example of the resultant placement using the PLA: a masculine-shaped line drawing.

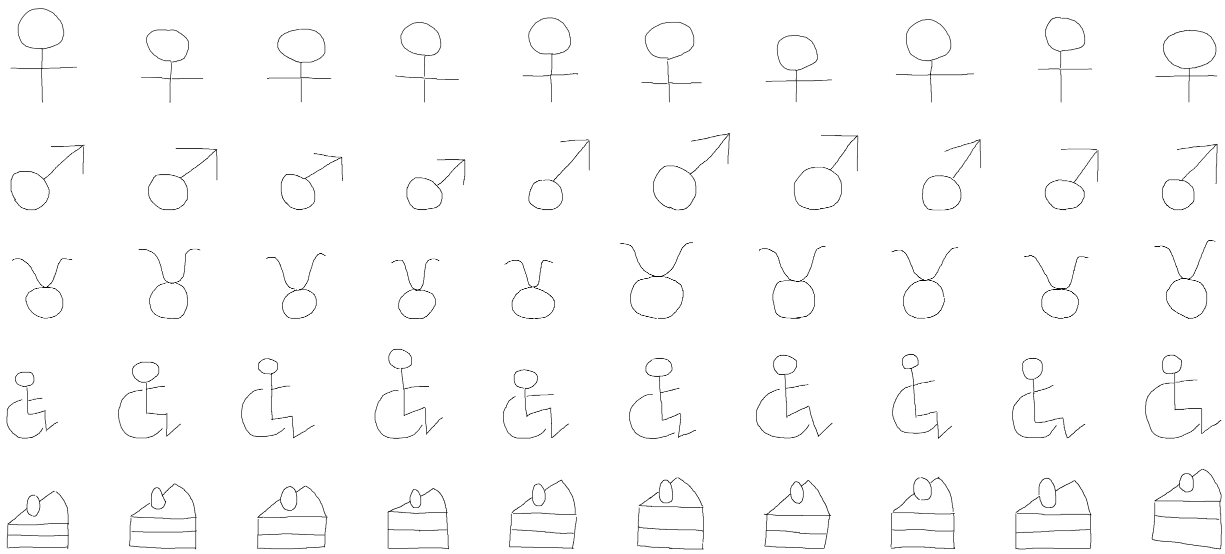

Figure 2. All line drawings for the five shape classes: feminine, masculine, Taurus, wheelchair, and cake.

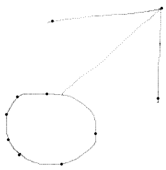

Figure 6. Example of the resultant placement using our method: a masculine-shaped line drawing.

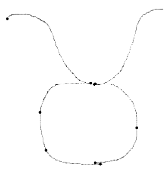

Figure 7. Example of the resultant placement using the PLA: a Taurus-shaped line drawing.

3.2 Bullseye test

The bullseye test is sometimes used in shape retrieval (e.g., see [12, 11, 5]). According to this test, each line drawing is chosen as a query and matched against all line drawings. The number of correct matches in the top 10 matches is counted by checking the line drawings found in the first 10 most similar matches for the matching cost with the vMDs. Because there are 10 line drawings in each class, the number of correct matches is at most 10×50 when all the line drawings have been chosen as queries. Thus, dividing the number of correct matches by 10×50 gives the overall retrieval rate for the top 10. The retrieval rate for the top five is given likewise. The effectiveness of the PLA and our method is then evaluated in terms of the retrieval rates.

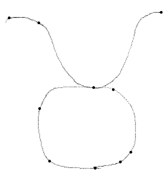

Figure 8. Example of the resultant placement using our method: a Taurus-shaped line drawing.

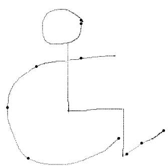

Figure 9. Example of the resultant placement using the PLA: a wheelchair-shaped line drawing.

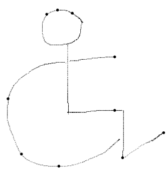

Figure 10. Example of the resultant placement using our method: a wheelchair-shaped line drawing.

3.3 Results

Figures 3–12 show examples of the resultant placements using the PLA presented in [1, 2] and our method with edge width $w = 11$. The thick crosses in

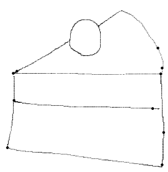

Figure 11. Example of the resultant placement using the PLA: a cake-shaped line drawing.

Table 1. Relation between the number of points and runtime for processing the points.

	Feminine in Figs. 3 & 4	Masculine in Figs. 5 & 6	Taurus in Figs. 7 & 8	Wheelchair in Figs. 9 & 10	Cake in Figs. 11 & 12
No. of points on the contours	1073	1010	1161	1657	3458
PLA	6365 (s)	6120 (s)	6844 (s)	9361 (s)	21601 (s)
Our method	3730 (s)	1802 (s)	1883 (s)	2958 (s)	19929 (s)

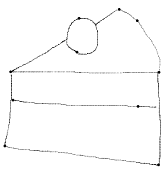

Figure 12. Example of the resultant placement using our method: a cake-shaped line drawing.

Table 2. Retrieval rates.

No. of top matches	PLA	Our method
Top five matches	.792	.924
Top 10 matches	.63	.808

the figures represent sample points on each line drawing. We observe from the figures that compared with the PLA, our method placed sample points on the characteristic locations of contours; thus, the sample points represented their shapes well. Moreover, Table 1 presents the number of points in these examples and the runtime for obtaining the placement of sample points. The runtime was measured on a personal computer with a 3.5GHz 6-Core Intel Xeon E5 processor and macOS X 10.10. Both the PLA and our method were implemented in R. The runtimes of both methods were in proportion to the number of sample points. Clearly, the runtime of our method was less than that of the PLA. The PLA is based on dynamic programming that requires a heavy computational cost for the number of points on the contours, whereas our method is not based on dynamic programming and the actual runtime depends on the selection of the initial vertices.

Next, Table 2 presents the retrieval rate for the top five and top 10 matches. We confirm from the table that our method yielded more effective placement in terms of shape retrieval than the PLA. Although there exists no certain stroke order and number in these line drawings, the PLA relies on such an order and number that are not invariant in the line drawings. As a result, it failed to place sample points at good locations on the shape. In contrast to the PLA, our method does not depend on the stroke order and number. Thus, our method places sample points at good locations even when the line drawings in each class are given by a different stroke order, stroke number, and shape scale.

4 Summary

In this paper, we presented an optimization method for placing sample points suitably on contours given by an uncertain stroke order and number. Our method is based on the assumption that each sample point should be located at a point that has a high degree and long edge. Using the various shapes of line drawings drawn using a different stroke order, stroke number, and scale, we demonstrated that our method is more effective in terms of shape retrieval than the PLA.

Acknowledgments

This work was supported in part by JSPS KAKENHI Grant Number 16K00308.

References

[1] J.-C. Perez and E. Vidal, "An algorithm for the optimum piecewise linear approximation of digitized curves," in *Proceedings of the 11th International Conference on Pattern Recognition*, vol. III, IAPR. The Hague, Netherlands: IEEE, Aug. 1992, pp. 167–170.

[2] ——, "Optimum polygonal approximation of digitized curves," *Pattern Recognition Letters*, vol. 15, no. 8, pp. 743–750, Aug. 1994.

[3] M. Cui, J. Femiani, J. Hu, P. Wonka, and A. Razdan, "Curve matching for open 2D curves," *Pattern Recognition Letters*, vol. 30, no. 1, pp. 1–10, Jan. 2009.

[4] L. Younes, *Shapes and Diffeomorphisms*, ser. Applied mathematical sciences. Berlin: Springer, 2010.

[5] H. Fan, Y. Cong, and Y. Tang, "Self-closed partial shape descriptor for shape retrieval," in *Proceedings of the 19th IEEE International Conference on Image Processing*. Orlando, FL: IEEE, Sep. 2012, pp. 505–508.

[6] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by direct search: New perspectives on some classical and modern methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, 2003.

[7] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*. Upper Saddle River, NJ: Pearson Education, 2004.

[8] R. J. Wilson, *Introduction to Graph Theory*, 4th ed. Harlow, Longman: Pearson Education, 1996.

[9] K. Iwata, "Drawing panel," Website, Oct. 2013. [Online]. Available: http://www.prl.info.hiroshima-cu.ac.jp/~kiwata/panel/

[10] K. Ose and K. Iwata, "Data set of line drawings with different stroke order and number," Website, May 2017. [Online]. Available: http://www.prl.info.hiroshima-cu.ac.jp/~kiwata/oseiwata/

[11] K. Ueaoki, K. Iwata, N. Suematsu, and A. Hayashi, "Matching handwritten line drawings with von Mises distributions," *IEICE Transactions on Information and Systems*, vol. E94-D, no. 12, pp. 2487–2494, Dec. 2011.

[12] C. Grigorescu and N. Petkov, "Distance sets for shape filters and shape recognition," *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1274–1286, Oct. 2003.