2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Reducing the Computational Cost of Shape Matching with the Distance Set

Kazunori Iwata
Graduate School of Information Sciences
Hiroshima City University
Hiroshima 731-3194, Japan
Email: kiwata@hiroshima-cu.ac.jp

***Abstract*—The distance set is known to be a versatile local descriptor of shape. As this is simply a set of ordinary distances between sample points on a shape, it is easy to construct and use. More importantly, it remains invariant under many settings and deformations, unlike other typical descriptors. However, in shape matching with distance sets, there is a tradeoff between performance and computational feasibility. In this paper, we present a new descriptor by improving the choice and order of elements in the distance set. We show that our descriptor is more efficient for shape matching from the viewpoint of computer algorithms. Additionally, we demonstrate that, although our descriptor runs more quickly in practice, it is equivalent to the original distance set in terms of shape retrieval.**

***Index Terms*—distance set, local descriptor of shape, shape matching, Hilbert distance**

I. Introduction

A shape is usually drawn with line drawings, figures, or object contours. For a reference point on a shape, a local descriptor is a means of expressing part of the shape around that point. Local descriptors are used to define a matching cost function that quantifies the dissimilarity between shapes. Thus, shape matching is strongly reliant on the local descriptor, and so a number of studies have examined various descriptors [1], [2], [3], [4], [5], [6], [7], [8], [9]. From a practical point of view, the distance set [2] is the most versatile local descriptor. The distance set around a reference point is simple to construct and easy to implement, because it is simply a set of ordinary distances between a given point and several nearby points. Consequently, it is invariant under isometric operations such as translation, rotation, and reflection. In addition, the distance set remains unaffected by whether a shape is represented with an ordered set of points or an unordered set. Furthermore, from the viewpoint of computer algorithms, the distance set is worth revisiting for practitioners. This is because it is essentially based on the sorting of distances, which is a well-studied problem in computer algorithms. However, the computational feasibility of shape matching can be an issue when using the distance set. In shape matching, the basic operation is the search. Generally, the larger the number of distances in the set, the better the resulting performance. However, the number of searches grows factorially with respect to the number of distances. This problem becomes serious as the number of distances increases. Thus, there is a tradeoff between performance and computational feasibility. Based on the above, the purpose of this paper is to address the problem by reducing the factorial growth. We thus present a new descriptor that is also a set of ordinary distances. The difference between the original distance set and our descriptor is simply in the choice and order of the distances. We show that our descriptor is more computationally efficient for shape matching, and, using a dataset of line drawings, demonstrate that it is equivalent to the distance set in terms of shape retrieval.

This paper is organized as follows. We examine the distance set in Section II. Based on this examination, we present a new local descriptor in Section III. Using a dataset of line drawings, we then present several experimental results in Section IV. Finally, we summarize the main points of this paper in Section V.

II. Preliminaries

A finite set of points is usually constructed by selecting some points on the shape of interest, known as sample points. The set of sample points is called the sample set. Shape matching, which is fundamental in shape recognition, is formulated in terms of determining the correspondence between sample sets.

A. Shape Matching and Dissimilarity

We describe a unified formulation of shape matching. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be shapes, and $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$ be their respective sample sets. Shape matching generally aims to find the correspondence between these sample sets. We represent the correspondence as a many-to-one map $M : \hat{\mathcal{S}}_1 \to \hat{\mathcal{S}}_2$, and denote the set of correspondences by

$$\mathscr{M} = \left\{ M \,|\, M : \hat{\mathcal{S}}_1 \to \hat{\mathcal{S}}_2 \right\}. \tag{1}$$

Given a matching cost function (MCF) $C : \hat{\mathcal{S}}_1 \times \hat{\mathcal{S}}_2 \to \mathbb{R}$, the optimal correspondence in terms of C is described by

$$M^* = \operatorname*{argmin}_{M \in \mathscr{M}} \frac{1}{\left|\hat{\mathcal{S}}_1\right|} \sum_{p \in \hat{\mathcal{S}}_1} C\left(p, M(p)\right). \tag{2}$$

Instinctively, the MCF is designed to express how different the distribution of sample points around p in $\hat{\mathcal{S}}_1$ is from that around $M(p)$ in $\hat{\mathcal{S}}_2$. Using the optimal correspondence, the dissimilarity between $\mathcal{S}_1$ and $\mathcal{S}_2$ is measured by

$$d\left(\mathcal{S}_1, \mathcal{S}_2\right) = \frac{1}{\left|\hat{\mathcal{S}}_1\right|} \sum_{p \in \hat{\mathcal{S}}_1} C\left(p, M^*(p)\right) + \sigma\left(\left|\hat{\mathcal{S}}_1\right|, \left|\hat{\mathcal{S}}_2\right|\right), \tag{3}$$

978-1-5090-4846-5/16/$31.00 ©2016 IEEE

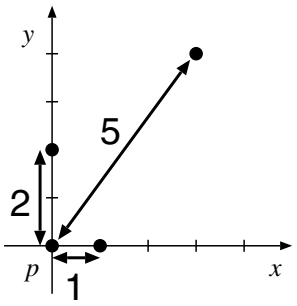

Fig. 1. Elements in $\hat{\mathcal{S}}$ and $p = (0, 0)$.

where σ is a function that tends to return a large real number as the difference between $\left|\hat{\mathcal{S}}_1\right|$ and $\left|\hat{\mathcal{S}}_2\right|$ increases. The dissimilarity between the shapes is the essence of shape recognition.

B. Distance Set

A sample point on the shape to which we are referring is called a reference point. In fact, the distance set (DS) [2] represents the distribution of sample points around a reference point as the set of distances between that point and some others. If the number of elements in the set is fixed to n, the set is called the n-DS. For any shape $\mathcal{S}$, the n-DS around $p \in \hat{\mathcal{S}}$ is described by

$$s_n(p) \stackrel{\text{def}}{=} \{l_1(p), \ldots, l_n(p)\}, \tag{4}$$

where $l_i(p)$ denotes the Euclidean distance between p and the i-th nearest point in $\hat{\mathcal{S}}$, that is,

$$l_1(p) \leq \cdots \leq l_n(p). \tag{5}$$

For example, if $\hat{\mathcal{S}} = \{(0, 0), (1, 0), (0, 2), (3, 4)\}$ and $p = (0, 0)$ in the x-y plane, as shown in Figure 1, then the 1-DS, 2-DS, and 3-DS around p are $s_1(p) = \{1\}$, $s_2(p) = \{1, 2\}$, and $s_3 = \{1, 2, 5\}$, respectively. For all $p \in \hat{\mathcal{S}}_1$ and $q \in \hat{\mathcal{S}}_2$, the MCF of the n-DS is defined by

$$C_{\mathrm{DS}}(p, q) \stackrel{\text{def}}{=} \min\left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{\left|l_i(p) - l_{\phi(i)}(q)\right|}{\max\left\{l_i(p), l_{\phi(i)}(q)\right\}} \;\middle|\; \phi \in \Phi_n \right\}, \tag{6}$$

where $l_i(p) \in s_n(p)$ and $l_{\phi(i)}(q) \in s_n(q)$ for all i, and Φ_n is the set of injections from $\{1, \ldots, n\}$ to $\{1, \ldots, n\}$.

C. Importance of Injection in the MCF

We have explained the matching cost between the DSs. Whether the matching cost is provided by an injection is important. For example, assume $s_2(p) = \{1, 2\}$, $s_2(q) = \{2, 3\}$, and $s_2(r) = \{2, 30\}$, as illustrated in Figure 2. Each dot in the figure denotes a sample point. In the figure, it appears that p is more similar to q than r. This is supported by the injection. In fact,

$$C_{\mathrm{DS}}(p, q) < C_{\mathrm{DS}}(p, r), \tag{7}$$

because their minimal costs are

$$C_{\mathrm{DS}}(p, q) = \frac{1}{2}\left(\frac{|1 - 3|}{\max\{1, 3\}} + \frac{|2 - 2|}{\max\{2, 2\}}\right) = \frac{1}{3}, \tag{8}$$

and

$$C_{\mathrm{DS}}(p, r) = \frac{1}{2}\left(\frac{|1 - 30|}{\max\{1, 30\}} + \frac{|2 - 2|}{\max\{2, 2\}}\right) = \frac{29}{60}, \tag{9}$$

respectively. However, if the matching cost is provided by an ordinary many-to-one map, rather than an injection, then the inequality does not hold, because, in this case, $C_{\mathrm{DS}}(p, q) = 1/3$ and

$$C_{\mathrm{DS}}(p, r) = \frac{1}{2}\left(\frac{|1 - 2|}{\max\{1, 2\}} + \frac{|2 - 2|}{\max\{2, 2\}}\right) = \frac{1}{4}. \tag{10}$$

Thus, whether the matching cost is provided by an injection plays a vital role in the MCF of the DS.

D. Merits and Demerits of the DS

Before moving to the main results, it is worthwhile clarifying the merits and demerits of the DS.

There are three notable merits. First, the DS around a reference point is simple to construct and easy to implement, because it is solely a set of ordinary distances between the given point and several nearby points. Thus, we need only sort the distances (a nearest point search), which is a well-studied problem in computer science, when forming the DS around that point. Second, as the DS consists only of Euclidean distances, it is clearly invariant under isometric operations such as translation, rotation, and reflection. Third, the DS remains unaffected by whether a shape is represented with an ordered set of points or an unordered set. Compared with other typical descriptors [1], [3], [4], [5], [6], [7], [8], [9], the DS remains invariant for many settings and deformations. When we consider the practical applications of the DS, these merits will be advantageous.

The disadvantage of the DS is the computational cost of the MCF. Generally, the performance of the n-DS increases with n, as will be demonstrated later. Accordingly, we wish to set n to be a large number. The minimization in (6) searches for the best injection from all the $n!$ injections in Φ_n. Hence, the number of searches grows factorially with n. Consequently, the MCF tends to be computationally infeasible as n increases, and thus there is a tradeoff between performance and computational feasibility. The computational cost of the minimization operation is a serious problem, and the focus of this paper is to reduce it.

III. Main Results

We start by considering the computational complexity of matching with the n-DS. Incidentally, little attention was given to the computational complexity in the original paper [2]. In the following, we use O-notation; refer to [10] for details.

A. Complexity of the DS

Let the number of elements in the sample set of a shape be m. The simplest algorithm for constructing the n-DSs for all sample points can be described as follows: 1) Form the $m \times m$ distance matrix whose components are the Euclidean distances between two sample points, and 2) find the n sample points nearest to each sample point. Forming the distance

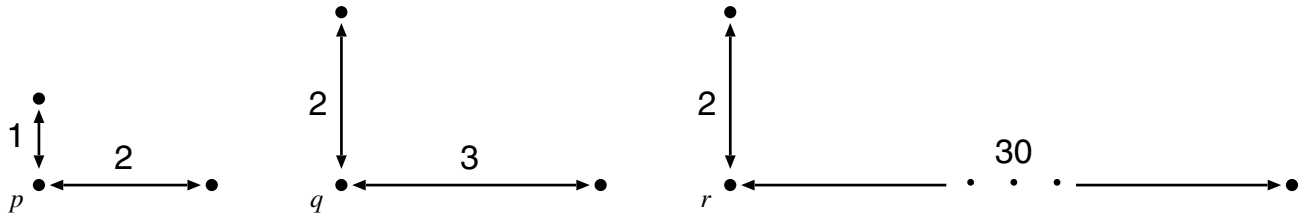

Fig. 2. Distance sets around p, q, and r.

matrix takes $O\left(m^2\right)$ time. Finding the n sample points nearest to each sample point takes $O(m \log m)$, because finding these essentially corresponds to sorting the distances. Thus, the total time required to construct the n-DSs for all sample points is $O(m^2 \log m)$.

Next, the simplest algorithm for matching two shapes with the MCF should be to search for the best injection. The computation time of the MCF in (6) is $O(n!n)$, because the summation takes $O(n)$ time and there are $n!$ injections in Φ_n. The total time required to match two shapes having m sample points with the MCF is $O(m^2 n!n)$.

Table I summarizes the time complexity of the n-DS. We have seen that making all the n-DSs takes $O(m^2 \log m)$ time and matching two shapes takes $O(m^2 n!n)$ time. Clearly, the $O(m^2 n!n)$ time is the more serious issue.

B. A New Descriptor

We describe a new criterion to reduce the $O(m^2 n!n)$ time. We deal with the convex hull of the sample set of a shape. Consider two sample points a and b in the sample set. We assume that the region enclosed by the convex hull is bounded. Then, the line through the two sample points intersects the convex hull at two points. Let u and v denote the intersection points, illustrated in Figure 3. The shaded portion in the figure denotes the region enclosed by the convex hull. For any real number $\delta > 0$, we define the distance between a and b as

$$d\,(a, b\,;\,\delta) \stackrel{\text{def}}{=} \left|\log \frac{(\overline{ua}+\delta)(\overline{vb}+\delta)}{(\overline{ub}+\delta)(\overline{va}+\delta)}\right|, \tag{11}$$

where $\overline{ua}$, for example, denotes the Euclidean distance between u and a. We call this distance the δ-Hilbert distance, because the limiting distance $\lim_{\delta\to 0} d\,(a, b\,;\,\delta)$ can be seen in the work of David Hilbert [11], [12]. Our distance can be regarded as an extension of his distance metric. If we allow δ to be zero in (11), the fraction with $\delta = 0$ denotes the cross ratio [13], [14] of u, a, b, and v.

We now explain the geometrical meaning of the δ-Hilbert distance. Consider two circles centered on u and v with a radius of δ. Let u', v' denote the outer intersection points where the two circles cut the line through the sample points a and b (see Figure 3). Using these points, (11) can be simplified to

$$d\,(a, b\,;\,\delta) = \left|\log \frac{\overline{u'a}\cdot\overline{v'b}}{\overline{u'b}\cdot\overline{v'a}}\right|. \tag{12}$$

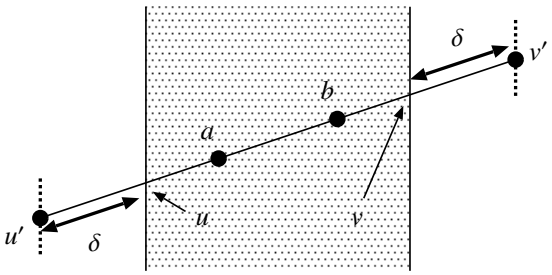

Fig. 3. Convex hull of a sample set.

The fraction is again the cross ratio of the four points. As a result, setting $\delta > 0$ enlarges the convex hull by a margin of δ. Henceforth, we call δ the margin width.

We now define our descriptor with the δ-Hilbert distance. Just as for the DS in (4), our descriptor is a set of Euclidean distances between a reference point and other sample points. The difference between the DS and our descriptor lies in the choice and order of the distances. Whereas the DS sorts the distances in ascending order, our descriptor sorts the distances in ascending order of their corresponding δ-Hilbert distances. That is, for any shape $\mathcal{S}$ and reference point $p \in \hat{\mathcal{S}}$, our descriptor, called the n δ-Hilbert distance set (n-HDS), is described by

$$s'_n(p\,;\,\delta) \stackrel{\text{def}}{=} \{l'_1(p), \ldots, l'_n(p)\}, \tag{13}$$

where n is the number of elements in the set, and $l'_i(p)$ denotes the Euclidean distance between p and the i-th nearest point in $\hat{\mathcal{S}}$ in terms of the δ-Hilbert distance. Consequently, the nearest sample points vary with δ, and are not always the same as those in the DS. The first point to note is that the elements in the set are Euclidean distances, but not δ-Hilbert ones, and the second point is that the inequality,

$$l'_1(p) \le \cdots \le l'_n(p), \tag{14}$$

does not always hold. For all $p \in \hat{\mathcal{S}}_1$ and $q \in \hat{\mathcal{S}}_2$, the MCF for the n-HDS is defined by

$$C_{\mathrm{HDS}}(p, q) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \frac{|l_i(p) - l_i(q)|}{\max\{l_i(p), l_i(q)\}}, \tag{15}$$

where $l_i(p) \in s_n(p)$ and $l_i(q) \in s_n(q)$ hold for all i. This corresponds to a limited version of the MCF for the DS, because letting ϕ in (6) be the identity mapping immediately

TABLE I
TIME COMPLEXITY.

Procedure	Detail Steps	n-DS	n-HDS
making n-DS/HDS	detecting convex hull	–	$O(m \log m)$
	making δ-Hilbert distance matrix	–	$O\left(m^3\right)$
	making Euclidean distance matrix	$O\left(m^2\right)$	$O\left(m^2\right)$
	finding n nearest points	$O(m^2 \log m)$	$O(m^2 \log m)$
matching with n-DS/HDS	matching two shapes with the MCF	$O(m^2 n!n)$	$O(m^2 n)$

gives $C_{\mathrm{HDS}}(p, q)$. In (15), it is important that there is no minimization operation, unlike the MCF for the DS. Instead of the minimization, the HDS computes the matching cost by choosing an appropriate margin width for δ.

It is important to note the role of the sorting operation based on the δ-Hilbert distance. For the sample set of a shape, an internal sample point inside the convex hull tends to be one of the n nearest points to the other sample points. At the same time, a sample point on the convex hull tends not to be one of the n nearest points to the other sample points, particularly those sample points on another side of the convex hull. Intuitively, whereas the DS sorts the distances according to the relative locations of their points, the HDS sorts according to the absolute locations of the points in the convex hull.

C. Complexity of the HDS

We now consider the time complexity of the n-HDS. Let the number of elements in the sample set of a shape be m. The simplest algorithm for constructing the n-HDSs for all sample points can be described as follows: 1) Detect the convex hull of the sample set; 2) make the $m \times m$ δ-Hilbert and Euclidean distance matrices; and 3) find the n nearest sample points in terms of the δ-Hilbert distance to each sample point. The time required to form the convex hull of the sample set is $O(m \log m)$. Making the δ-Hilbert distance matrix takes $O\left(m^3\right)$ time, and the Euclidean distance matrix takes $O\left(m^2\right)$ time. Because finding the n nearest sample points in terms of the δ-Hilbert distance to each sample point takes $O(m \log m)$, the time required to construct the n-HDS for all sample points is $O(m^2 \log m)$. Overall, making the n-HDSs for all sample points runs in $O\left(m^3\right)$ time. This is slightly longer than required for the DS.

Next, the algorithm for matching two shapes with the MCF is solely the summation in the MCF. Thus, the computation time of the MCF in (15) is $O(n)$. When the two shapes have m sample points, the total matching time with the MCF is $O(m^2 n)$. This is somewhat less than for the DS. Table I summarizes the time complexity of the n-HDS. We can see from this table that, although the n-HDS takes slightly longer to make the set, it requires far less time to match shapes.

IV. COMPUTATIONAL EXPERIMENTS

We compared the DS and the HDS in terms of their shape retrieval rates and actual runtimes.

A. Dataset

We used the dataset available from [15] employed in [9]. This dataset consists of 200 single-stroke line drawings. All the line drawings fall into 10 shape classes, each of which contains 20 drawings. Figure 4 illustrates some of the single-stroke line drawings in the 10 classes.

The size of each line drawing was normalized by making its bounding box a fixed area of 40000, in keeping with the aspect ratio of the box. After this normalization, 15 sample points were uniformly extracted from each line drawing to form the sample set. As all the sample sets were the same size, the second term of the dissimilarity in (3) was zero.

B. Evaluation

Each line drawing was selected as a query and matched against all 200 line drawings. The number of correct matches in the top 20 matches was counted by examining the line drawings found in the first 20 most similar matches for dissimilarities. As every class has 20 line drawings, the total number of correct matches is at most 4000 when all the line drawings have been selected as queries. Thus, we obtained the overall retrieval rate for the top 20 matches by dividing the number of correct matches by 4000. The effectiveness of the local descriptors was then assessed in terms of the retrieval rates. Such an evaluation is called a Bulls-eye test, and is often used in shape retrieval (see [2], [7], [9], for example).

C. Results

Figure 5 shows the retrieval rates for the n-DS and n-HDS with a margin of δ = 600. Table II shows the runtime (s) of the overall experiment, which consists of both making the n-DSs/HDSs and matching all 200 line drawings. We abandoned the experimental results for the 9-DS and above, because these took too long. The experiments were performed on a personal computer with an Intel Core i7-975 processor and Windows 7. Both the DS and HDS were implemented in R [16].

We can see from Table II that the runtime of the HDS is much less than that of the DS. This agrees with the asymptotic analysis in Table I. Incidentally, the 589946 and 1346 seconds in the case of n = 8 correspond approximately to a week and 22 minutes respectively. Of course, the runtime will depend on the specific computer environment.

Figure 5 shows that the performance of both n-DS and n-HDS improved as n increased. Somewhat surprisingly, we can see that the n-HDS is equivalent to n-DS in terms of shape

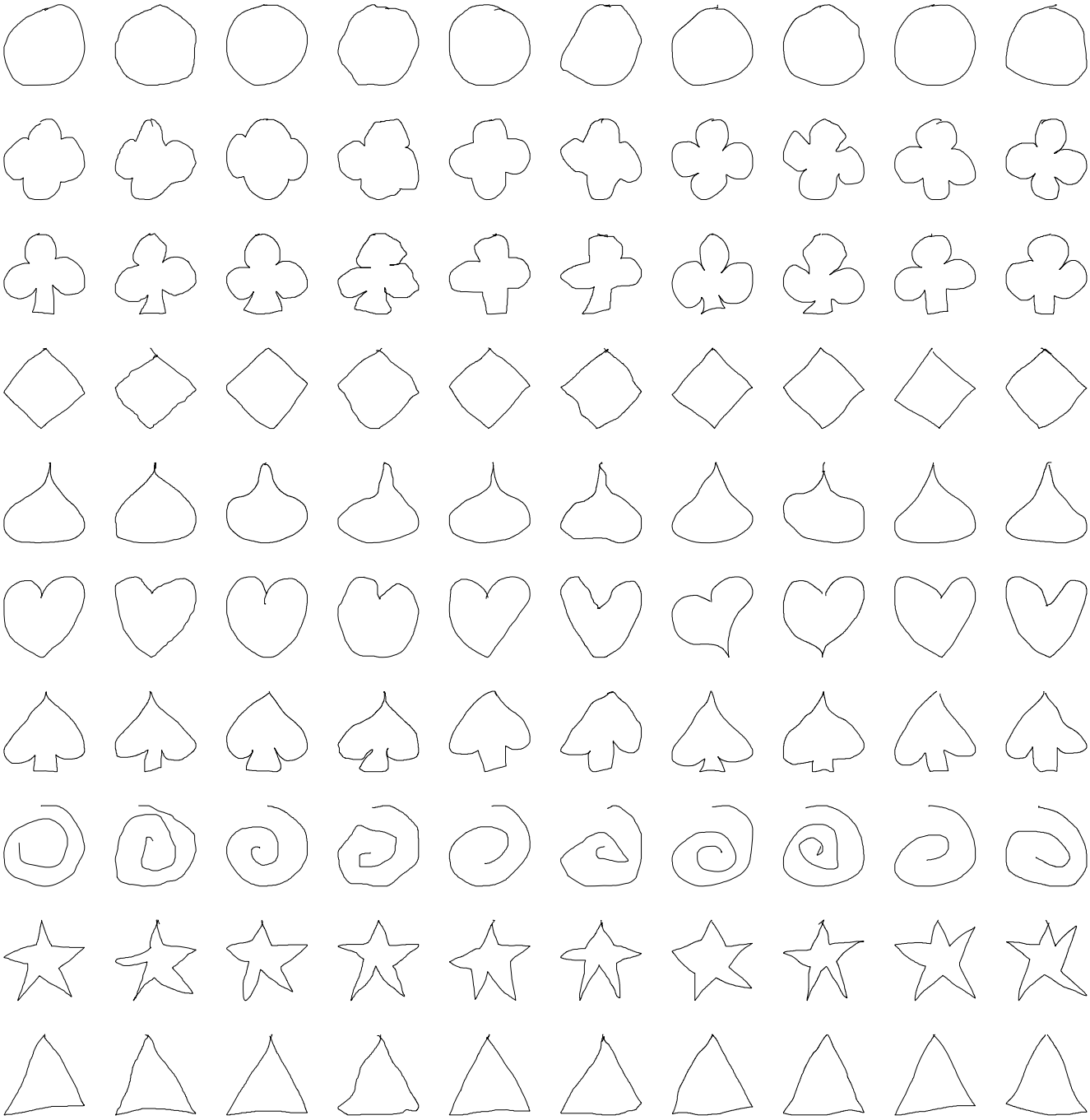

Fig. 4. Example single-stroke line drawings in the 10 classes: circle, cloud, club, diamond, drop, heart, spade, spiral, star, triangle.

retrieval. This implies that the matching cost based on the δ-Hilbert distance works well.

V. Summary

We have presented a new descriptor, called the HDS, using a matching cost based on the δ-Hilbert distance. Although the HDS takes a slightly longer time to construct than the DS, it is significantly faster at matching shapes. Using various line drawings, we demonstrated that, although our descriptor runs in a much shorter time, it is equivalent to the DS in terms of shape retrieval.

It is clear that the choice of a good margin width plays a vital role in shape matching. Thus, finding an efficient tuning mechanism for this margin width is an important area of future study.

Acknowledgment

We wish to thank Mr. Masaki Takasuga and Ms. Kana Ishihara, Hiroshima City University, for their support with the

TABLE II
RUNTIME (S).

n	1	2	3	4	5	6	7
n-DS	1011	1366	2220	4489	11673	38538	140281
n-HDS	1178	1240	1258	1275	1311	1329	1374
n	8	9	10	11	12	13	14
n-DS	589946	–	–	–	–	–	–
n-HDS	1346	1353	1373	1371	1376	1330	1350

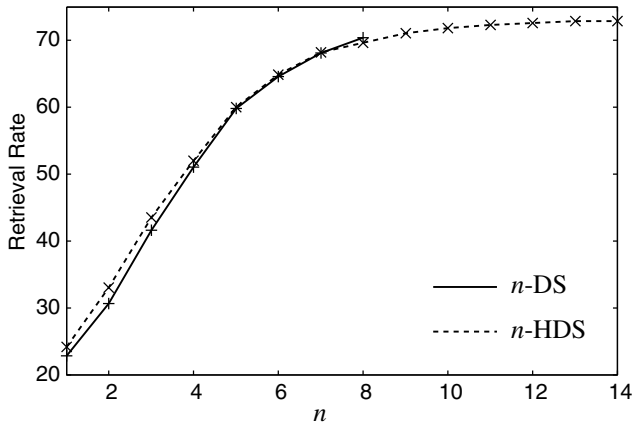

Fig. 5. Retrieval rate.

computational experiments.

REFERENCES

[1] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "On aligning curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 1, pp. 116–125, Jan. 2003.

[2] 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.

[3] S. Belongie, G. Mori, and J. Malik, "Matching with shape contexts," in *Statistics and Analysis of Shapes*, ser. Modeling and Simulation in Science, Engineering and Technology, H. Krim and A. Yezzi Jr., Eds. Boston: Birkhäuser, 2006, pp. 81–105.

[4] S. Manay, D. Cremers, B.-W. Hong, A. J. Yezzi Jr., and S. Soatto, "Integral invariants for shape matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1602–1618, Oct. 2006.

[5] 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.

[6] 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.

[7] 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.

[8] H. Chatbri, K. Davila, K. Kameyama, and R. Zanibbi, "Shape matching using keypoints extracted from both the foreground and the background of binary images," in *Proceedings of the 2015 International Conference on Image Processing Theory, Tools and Applications*, IEEE. Orleans, France: IEEE, Nov. 2015, pp. 205–210.

[9] K. Iwata, N. Suematsu, and A. Hayashi, "A new shape descriptor based on an angular-linear probability distribution," in *Proceedings of the 3rd IAPR Asian Conference on Pattern Recognition*, IAPR. Kuala Lumpur, Malaysia: IAPR, Nov. 2015, pp. PS2–35.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. Cambridge, MA: MIT Press, Jul. 2009.

[11] D. Hilbert, "Über die gerade linie als kürzeste verbindung zweier punkte," *Mathematische Annalen*, vol. 46, no. 1, pp. 91–96, Mar. 1895.

[12] ——, *The Foundations of Geometry*, 2nd ed. La Salle, Illinois: Open Court, 1971.

[13] R. Hartshorne, *Geometry: Euclid and Beyond.* New York: Springer, 2000.

[14] D. E. Blair, *Inversion Theory and Conformal Mapping*. Providence, RI: American Mathematical Society, 2000.

[15] M. Takasuga and K. Iwata, "Data set of single-stroke line drawings," Jun. 2015. [Online]. Available: http://www.prl.info.hiroshima-cu.ac.jp/~kiwata/takasugaiwata/

[16] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. [Online]. Available: https://www.R-project.org/