An Extended Scheme for Shape Matching with Local Descriptors

Kazunori IWATA\textsuperscript{1(a)}, Member, Hiroki YAMAMOTO\textsuperscript{1}, Nonmember, and Kazushi MIMURA\textsuperscript{1}, Member

SUMMARY Shape matching with local descriptors is an underlying scheme in shape analysis. We can visually confirm the matching results and also assess them for shape classification. Generally, shape matching is implemented by determining the correspondence between shapes that are represented by their respective sets of sampled points. Some matching methods have already been proposed; the main difference between them lies in their choice of matching cost function. This function measures the dissimilarity between the local distribution of sampled points around a focusing point of one shape and the local distribution of sampled points around a referring point of another shape. A local descriptor is used to describe the distribution of sampled points around the point of the shape. In this paper, we propose an extended scheme for shape matching that can compensate for errors in existing local descriptors. It is convenient for local descriptors to adopt our scheme because it does not require the local descriptors to be modified. The main idea of our scheme is to consider the correspondence of neighboring sampled points to a focusing point when determining the correspondence of the focusing point. This is useful because it increases the chance of finding a suitable correspondence. However, considering the correspondence of neighboring points causes a problem regarding computational feasibility, because there is a substantial increase in the number of possible correspondences that need to be considered in shape matching. We solve this problem using a branch-and-bound algorithm, for efficient approximation. Using several shape datasets, we demonstrate that our scheme yields a more suitable matching than the conventional scheme that does not consider the correspondence of neighboring sampled points, even though our scheme requires only a small increase in execution time.

key words: shape matching, local shape descriptor, probability density estimator, branch-and-bound algorithm

1. Introduction

Shape matching with local descriptors (LDs) refers to an underlying scheme in shape analysis, whose results are useful for shape classification. Regarding shape classification, in recent years there has been renewed interest in deep learning-based methods\cite{1,2,3}. Typically, deep learning-based methods require preprocessing to extract the features of a shape and construct a feature vector as an input signal to a large-scale neural network. Additionally, they require a large number of feature vectors and a long period of time to train the network before performing classification using the network. It is troublesome for human experts to elaborate on why the network reached a particular classification solution. In contrast to classification based on such a machine learning method, shape matching with LDs is quite simple to perform and does not require preprocessing, such as creating feature vectors and the training period associated with them. In addition, because the matching results can be understood visually by human experts, it is possible to explain why the classifier arrived at a particular solution, by checking the matching results. Thus, shape matching with LDs is a type of explainable artificial intelligence, which has been discussed recently in\cite{4,5,6}.

A shape is formed of contours, such as handwriting with a pen or boundaries obtained by image segmentation. When we ignore the stroke order and number of contours, the contours can be expressed as a finite set of points, using contour sampling. Shape matching is implemented by determining the correspondence between the sets of sample points of each shape. Although several matching methods have been studied in the past few decades\cite{7,8,9,10,11}, the main difference between them is their choice of matching cost function (MCF). The MCF measures the dissimilarity between the local distribution of sample points around a focusing point of one shape and the local distribution of sample points around a referring point of another shape. An LD of a shape at a sample point is usually used to describe the distribution of sample points that is available only around the specified sample point of the shape.

Presenting a new LD that is superior to existing LDs constitutes the mainstream of research in shape matching, but we are not concerned with this in this paper. We approach the improvement of shape matching from a different perspective. We propose an effective scheme for determining the correspondence between shapes that can compensate for errors in existing local descriptors without the need to modify them. The key point of our scheme is to consider the correspondence of neighboring sample points of a focusing sample point when determining the correspondence of the focusing sample point. Consequently, we can increase the chance of finding an appropriate correspondence. However, considering neighboring sample points causes another problem: there is a substantial increase in the number of correspondences that we have to check for matching. Our proposal for solving this problem is to introduce an efficient approximation to reduce the number of correspondences. Using several shape datasets, we demonstrate that our scheme yields a better correspondence than the conventional scheme that does not consider the correspondence of neighboring sample points, even though our scheme requires only a small increase in execution time.
The structure of this paper is as follows. We briefly describe the conventional scheme for matching in Sect. 2, and present our scheme, which is helpful for some LDs, in Sect. 3. Using several shape datasets, we compare the conventional scheme with our scheme in Sect. 4. Finally, we present the conclusions about our scheme in Sect. 5.

2. Shape Matching Scheme

We begin with the formulation of shape matching. We use $S$ to denote a shape formed of contours in the $x$-$y$ plane. Using contour sampling (for an example of sampling, see [12], [13]), the shape can be expressed as a finite set of points.

Let $\hat{S}$ be this finite set of points. Throughout this paper, such a finite set of points is called a sample set, and an element of a sample set is called a sample point. Thus, because shapes are expressed by their sample sets, shape matching involves determining the correspondence between the sample set of one shape and that of another.

Let $\mathbb{R}$ be the set of real numbers and $\mathbb{R}_+^*$ the set of nonnegative real numbers. Let $S_1$ and $S_2$ denote shapes, and $\hat{S}_1$ and $\hat{S}_2$ their respective sample sets. Assume that the correspondence between the shapes is expressed by a map: $M: \hat{S}_1 \rightarrow \hat{S}_2$. According to [7], [10], given an MCF $C: \hat{S}_1 \times \hat{S}_2 \rightarrow \mathbb{R}_+^*$, the correspondence in terms of $C$ is defined as

$$ \hat{M} = \arg \min_{M \in M(\hat{S}_1, \hat{S}_2)} \frac{1}{|\hat{S}_1|} \sum_{p \in \hat{S}_1} C(p, M(p)), $$

where $M(\hat{S}_1, \hat{S}_2)$ is the set of maps from $\hat{S}_1$ to $\hat{S}_2$, and $|\hat{S}_1|$ is the number of elements in $\hat{S}_1$. Intuitively, the MCF quantifies the dissimilarity between the local distribution of sample points around $p$ in $\hat{S}_1$ and the local distribution of sample points around $M(p)$ in $\hat{S}_2$. Regarding the correspondence $\hat{M}$, the matching cost between the shapes is calculated as

$$ d(S_1, S_2; \hat{M}) = \frac{1}{|\hat{S}_1|} \sum_{p \in \hat{S}_1} C(\hat{p}, \hat{M}(p)) + \sigma(\hat{S}_1, \hat{S}_2), $$

where $\sigma$ is a function such that the larger the difference between $|\hat{S}_1|$ and $|\hat{S}_2|$, the larger the value of $\sigma$. We describe the concrete MCFs of several LDs in Sect. 4.1. Constructing $\hat{M}$ in (1) has a relatively small computational cost. For any $p \in \hat{S}_1$, we obtain

$$ \hat{q} = \arg \min_{q \in \hat{S}_2} C(p, q), $$

and construct $\hat{M}$ such that $\hat{q} = \hat{M}(p)$. Hence, obtaining $\hat{M}$ involves computing $C(p, q) |\hat{S}_1||\hat{S}_2|$ times.

3. Proposed Matching Scheme

In this section, we extend the correspondence so that it depends not only on $p$ but also on the neighboring sample points of $p$.

3.1 Two Types of Probability Density Estimator

Let $r_1^{(k)}$ be the Euclidean distance between $p \in \hat{S}_1$ and the point in $\hat{S}_2$ that is the $k$-th nearest to $p$. Using $r_1^{(k)}$, we can define the $k$ nearest neighboring points of $p$ as

$$ N_1(p; k) = \{q \in \hat{S}_1 \mid \|p - q\| \leq r_1^{(k)}\}, $$

where $\| \cdot \|$ denotes the Euclidean norm. Similarly, let $r_2^{(k)}$ be the Euclidean distance between $M(p) \in \hat{S}_2$ and the point in $\hat{S}_2$ that is the $k$-th nearest to $M(p)$, and define the $k$ nearest neighboring points of $M(p)$ as

$$ N_2(M(p); k) = \{q \in \hat{S}_2 \mid \|M(p) - q\| \leq r_2^{(k)}\}. $$

Based on these two sets, we provide their respective probability densities using the $k$ nearest neighboring estimates [14], [15], as follows: for all $i \in \{1, 2\}$,

$$ \rho_i(p; k, M) = \frac{k}{|\hat{S}_i| V_i(M(p); k)}, $$

where $V_1(M(p); k)$ and $V_2(M(p); k)$ denote the volume of the minimum circles centered at $M(p)$ that contain all the points in $M(N_1(p; k))$ and $N_2(M(p); k)$, respectively, where

$$ M(N_1(p; k)) = \{M(q) \in \hat{S}_2 \mid q \in N_1(p; k)\}. $$

This is a set of sample points in $\hat{S}_2$ that are mapped by $M$ from $N_1(p; k)$. In fact, $V_1(M(p); k)$ and $V_2(M(p); k)$ are the areas defined by

$$ V_1(M(p); k) = \pi \max_{q \in M(N_1(p; k))} \|M(p) - q\|^2, $$

$$ V_2(M(p); k) = \pi \left(\frac{r_2^{(k)}}{2}\right)^2, $$

respectively.

Figure 1 illustrates an example of matching two shapes with a map $M$ in the case of $k = 3$. In the figure, the dots represent the sample points of shapes and the arrows represent the map from the sample set of one shape (on the left-hand side) to that of another shape (on the right-hand side). Let $p$ be the red point of the shape on the left-hand side, and let $M(p)$ be the red point on the right-hand side. Then, $N_1(p; 3)$ in (4) is depicted as the set of dots surrounded by the dashed line, and $M(N_1(p; 3))$ in (7) is depicted as the
set of dots to which the blue arrows point. There are two solid-line circles centered at \( M(p) \) on the right-hand side. The smaller circle, with a green radius \( r_2 \), depicts the circle whose area is \( V_2(M(p); 3) \), and the larger circle, with radius max\( q \in M(N_1(p; 3)) \| M(p) - q \| \), depicts the circle whose area is \( V_1(M(p); 3) \). We can confirm from the figure that the difference in area between the two circles plays a vital role in measuring how the three points neighboring \( p \) remain close to \( M(p) \) when mapped by \( M \).

3.2 Scheme with Additional Cost

We present an extended scheme to determine a correspondence between shapes. The important point here is to incorporate the difference in area between the two circles into the matching cost of the two shapes. Given an MCF \( C : \hat{S}_1 \times \hat{S}_2 \to \mathbb{R}_+^* \), our correspondence is defined as

\[
\hat{M}^* = \arg \min_{M \in \mathcal{M}({\hat{S}_1}, {\hat{S}_2})} \frac{1}{|\hat{S}_1|} \sum_{p \in \hat{S}_1} \left[ C(p, M(p)) + \alpha \delta(p; k, M) \right],
\]

(10)

where \( \alpha \in \mathbb{R}_+^* \) is the mixing rate that weights the first and second terms in the parentheses. It is worth noting that \( \hat{M} = \hat{M}^* \) holds if \( \alpha = 0 \) in (10). The additional cost function \( \delta \) is defined by

\[
\delta(p; k, M) = \max \left\{ 0, 1 - \frac{\rho_1(p; k, M)}{\rho_2(p; k, M)} \right\},
\]

(11)

using the convention that \( \rho_1(p; k, M) / \rho_2(p; k, M) = \infty \) if \( \rho_2(p; k, M) = 0 \) holds. Using (10), we can define the dissimilarity based on \( \hat{M}^* \) as

\[
d(\hat{S}_1, \hat{S}_2; \hat{M}^*) = \frac{1}{|\hat{S}_1|} \sum_{p \in \hat{S}_1} \left[ C(p, \hat{M}^*(p)) + \alpha \delta(p; k, \hat{M}^*) \right] + \sigma(\hat{S}_1, \hat{S}_2).
\]

(12)

The density ratio \( \rho_1(p; k, M) / \rho_2(p; k, M) \) in (11) becomes small when the \( k \) nearest neighbors of \( p \) are mapped by \( M \) to be far away from \( M(p) \), and vice versa. Accordingly, if \( M(N_1(p; k)) \subset N_2(M(p); k) \) holds, then the density ratio is greater than one, and hence the additional cost is zero; otherwise, the additional cost takes a value between zero and one. Thus, we add the cost to preserve the neighboring points in matching shapes, to increase the chance of finding a good correspondence overall.

3.3 Computational Cost

As will be shown in the next section, our scheme sometimes improves the performance of shape matching. However, it causes another problem, related to computational feasibility. To determine \( M(p) \) for any sample point \( p \in \hat{S}_1 \), we require the correspondence of neighboring sample points of \( p \); that is, we require \( M(p') \) for all \( p' \in N_1(p; k) \). Similarly, to determine \( M(p'') \), we require \( M(p''') \) for all \( p''' \in N_1(p'; k) \), and so on. Thus, this straightforward computation involves checking all possible maps from \( \hat{S}_1 \) to \( \hat{S}_2 \). As a result, the computational cost of finding \( \hat{M}^* \) is substantial because the number of all possible maps is \(|S_1| \cdot |S_2| \cdot m_1 \cdot m_2 \), where \( m_1 = |\hat{S}_1| \) and \( m_2 = |\hat{S}_2| \). Our proposed solution to this problem is to introduce an approximation of \( \hat{M}^* \), by narrowing the candidates of the map for finding \( \hat{M}^* \) to

\[
\hat{M}(\hat{S}_1, \hat{S}_2; C, \epsilon) = \left\{ M \in \mathcal{M}(\hat{S}_1, \hat{S}_2) \mid \sum_{p \in \hat{S}_1} C(p, M(p)) \leq \sum_{p \in \hat{S}_1} C(p, \hat{M}(p)) + \epsilon \right\},
\]

(13)

for any tolerance \( \epsilon \in \mathbb{R}_+^* \). Each candidate map is one whose sum of matching costs, in terms of MCF, is close to that of \( \hat{M} \). It is important to note that this set satisfies

\[
\hat{M}(\hat{S}_1, \hat{S}_2; C, \epsilon) \subseteq \hat{M}(\hat{S}_1, \hat{S}_2),
\]

(14)

and it includes \( \hat{M} \) in (1). Only if \( \epsilon = 0 \). We can then define the approximation of \( \hat{M}^* \) as

\[
\hat{M}^* = \arg \min_{M \in \mathcal{M}(\hat{S}_1, \hat{S}_2; C, \epsilon)} \frac{1}{|\hat{S}_1|} \sum_{p \in \hat{S}_1} \left[ C(p, M(p)) + \alpha \delta(p; k, \hat{M}^*) \right]
\]

(15)

for any tolerance \( \epsilon \in \mathbb{R}_+^* \).

The summation in (15) consists of the MCF and additional cost. In many cases, the time complexity of the MCF of a specific LD is much greater than that of the additional cost. The additional cost derives from the \( k \) nearest neighboring points of \( p \) in \( \hat{S}_1 \) and those of \( M(p) \) in \( \hat{S}_2 \) in (4) and (5), respectively. If we apply the usual sorting operation, such as quicksort, then the average time complexity of the \( k \) nearest neighboring points of \( p \) in \( \hat{S}_1 \) is \( O(m_1 \log m_1) \), and that of the \( k \) nearest neighboring points of \( M(p) \) in \( \hat{S}_2 \) is \( O(m_2 \log m_2) \). We have used O-notation; refer to [16] for details.

3.4 Branch-and-Bound Method

Equation (15) is a combinatorial optimization problem whose goal is to find the best correspondence in the reduced set in (13). This problem can be solved efficiently using a branch-and-bound (BB) algorithm [17] because the inequality form in (13) is suitable for recursive expression, as follows. The reduced set corresponds to the set of candidate solutions in the BB algorithm. Now, we arrange the elements of \( \hat{S}_1 \) in order of decreasing matching costs with \( \hat{M} \) in (1). Suppose that, as a result, we obtain \( p^{(1)}, p^{(2)}, \ldots, p^{(m_1)} \in \hat{S}_1 \) in this order, where \( m_1 = |\hat{S}_1| \). The order implies

\[
C(p^{(1)}, \hat{M}(p^{(1)})) \geq C(p^{(2)}, \hat{M}(p^{(2)})) \geq \ldots \geq C(p^{(m_1)}, \hat{M}(p^{(m_1)})).
\]

(16)
This arrangement is intended to avoid unnecessary branching in the BB algorithm. Let \( \hat{S}_2 = \{q_1, q_2, \ldots, q_m\} \) for now. For \( i = 1, 2, \ldots, m_1 \), we recursively check combinations of \( p^{(i)} \) and elements of \( \hat{S}_2 \). These combinations are illustrated as a rooted tree in Fig. 2. In the figure, each path from the root to a leaf denotes a map. Specifically, if the combinations do not satisfy

\[
\sum_{j=1}^{i} C\left(p^{(i)}, M\left(p^{(i)}\right)\right) \leq \sum_{j=1}^{i} C\left(p^{(i)}, \hat{M}\left(p^{(i)}\right)\right) + \epsilon, \quad (17)
\]

at the \((i - 1)\)-th depth level in the rooted tree, then the corresponding branch is pruned, thereby eliminating candidate solutions that will not be optimal.

3.5 Parameter Settings

As shown in (13) and (15), our scheme relies on the number of neighboring sample points \( k \), mixing rate \( \alpha \), and tolerance \( \epsilon \). Although there is no perfect method to obtain values of these parameters that are suitable for an arbitrary shape dataset, there is an empirical method to tune them. First, we concentrate on tolerance, because the execution time depends heavily on it. For the experiments described in the following section, we set it such that the number of map candidates in (13) was approximately 100. If we need to accelerate our scheme, then we should reduce the number of map candidates. Second, we should tune the mixing rate and the number of neighboring sample points simultaneously. The mixing rate may be set to a fixed positive value, by reference to the mean of MCFs in (15). Because the additional cost is between zero and one, it is relatively easy to find a good value for the mixing rate. In general, the number of neighboring sample points is related to the number of sample points \( |\hat{S}| \) in a shape \( S \). This proposal for setting these parameters is an imperfect approach but often works well in practice.

4. Experiments

Using some LDs of various types, we examine whether our scheme for matching shapes is effective, with only a relatively small increase in execution time.

4.1 Local Descriptors

We tested our matching scheme using four typical LDs that are applicable to shape contours represented as finite sets of points. The local descriptors are classified into three types: distance based, distance-and-angle based, and angle based. Our scheme is applicable to these LDs without any modification.

(1) Distance-Based LD

The distance set (DS) [7] is one of the simplest descriptors: it relies only on the distance between sample points of a shape. It represents the local distribution, around a sample point, as a set of distances between that point and others. If the number of elements in the set is fixed as \( n \), the set is called the \( n \)-DS. The \( n \)-DS around \( p \in \hat{S} \) is described by \( s_n(p) = \{l_1(p), \ldots, l_n(p)\} \), where \( l_i(p) \) denotes the Euclidean distance between \( p \) and the \( i \)-th nearest sample point in \( \hat{S} \). For example, consider the sample set \( \hat{S} = \{(0, 0), (0, 1), (1, 1), (3, 1), (2, 2)\} \). If the origin is the focusing point \( p \in \hat{S} \), then its 3-DS is \( s_3(p) = \{1, \sqrt{2}, 2, \sqrt{2}\} \).

For all \( p \in \hat{S}_1 \) and \( q \in \hat{S}_2 \), the MCF for the \( n \)-DS is defined by

\[
C_{DS}(p, q) = \min \left\{ \frac{1}{n} \sum_{i=1}^{n} \frac{|l_i(p) - l_{\phi(i)}(q)|}{\max\{l_i(p), l_{\phi(i)}(q)\}} : \phi \in \Phi_n \right\}, \quad (18)
\]

where \( l_i(p) \in s_n(p) \) and \( l_{\phi(i)}(q) \in s_n(q) \) hold for all \( i \) and

\[
\Phi_n = \{\phi : [1, \ldots, n] \rightarrow [1, \ldots, n], \forall i, j \phi(i) = \phi(j) \Rightarrow i = j\}, \quad (19)
\]

where \( \Rightarrow \) denotes logical implication, and hence \( \phi \) is an injection from \([1, \ldots, n]\) to \([1, \ldots, n]\). Thus, the DS is invariant to isometric transformation, such as translation, rotation, and reflection, but is distance-dependent.

As for parameter setting, the performance of the \( n \)-DS increases with \( n \) [18]. However, the minimization problem in (18) tends to be computationally infeasible as \( n \) increases. Thus, there is a trade-off between performance and computational feasibility.

(2) Distance-and-Angle-Based LD

The shape context (SC) [9] around a sample point depends not only on the distance between that point and another but also on the angle of deviation between the x-axis and the vector formed by the two sample points. The local distribution around a sample point is expressed as a histogram. Let \( B \) denote the total number of bins. The bins are sections drawn by dividing a disk by several concentric circles centered at a focusing sample point and several radials from that point. For all \( b \in [1, \ldots, B] \), the SC around \( p \in \hat{S} \) with respect to the \( b \)-th bin of the histogram is described as

\[
h_p(b) = \left| \{p' \in \hat{S}\cap[p] \mid p' - p \in \text{bin}(b)\} \right|, \quad (20)
\]
respectively. Then, \( \text{cant circular distribution in directional statistics} \)


tetration parameter, respectively. The mixture of vMDs

(3) Angle-Based LDs

Fig. 3 Example of bins of the SC in the case of \( B = 9 \).

where \( \setminus \) denotes set difference and \( \text{bin}(b) \) is the \( b \)-th bin. It is important to note that \( p' - p \) is a position vector quantity, with magnitude and direction. For intelligibility, we illustrate an example of bins in Fig. 3. In the figure, the dots depict the sample points, and the sections annotated with \( 1, \ldots, 9 \) depict the bins. If the red point is the focusing point \( p \), then \( h_p(1) = 1, h_p(2) = 0, h_p(3) = 1, h_p(4) = 2, h_p(5) = 1, h_p(6) = 2, h_p(7) = 2, h_p(8) = 4, \) and \( h_p(9) = 0 \).

For all \( p \in S_1 \) and \( q \in S_2 \), the MCF for the SC is defined by

\[
CSC(p, q) = \frac{1}{2} \sum_{b=1}^{B} \frac{\left( \tilde{h}_p(b) - \tilde{h}_q(b) \right)^2}{\tilde{h}_p(b) + \tilde{h}_q(b)}, \tag{21}
\]

with the convention that \( \left( \tilde{h}_p(b) - \tilde{h}_q(b) \right)^2 / \left( \tilde{h}_p(b) + \tilde{h}_q(b) \right) = 0 \) if \( \tilde{h}_p(b) + \tilde{h}_q(b) = 0 \) holds, where \( \tilde{h}_p(b) \) and \( \tilde{h}_q(b) \) are defined by

\[
\tilde{h}_p(b) = \frac{h_p(b)}{\sum_{b=1}^{B} h_p(b)} \quad \text{and} \quad \tilde{h}_q(b) = \frac{h_q(b)}{\sum_{b=1}^{B} h_q(b)}, \tag{22}
\]

respectively. Because the SC is dependent on both distance and angle, it is clearly more informative than the DS, which is dependent only on distance. However, it is not invariant to isometric transformation. As for parameter setting, to fix the bins, we need the number of circles, the radii of the circles, and the number of radiads; we denote these by \( \psi, \Psi, \) and \( \eta \), respectively. Then, \( B = \psi \times \eta \) holds for any \( \psi \) and \( \eta \) that are positive integers.

(3) Angle-Based LDs

The von Mises distribution (vMD) [19] is the most significant circular distribution in directional statistics [20]. For all angular variables \( \theta \), it is defined by

\[
f(\theta; \mu, \kappa) = \frac{\exp(\kappa \cos(\theta - \mu))}{\int_{0}^{2\pi} \exp(\kappa \cos \theta) d\theta}, \tag{23}
\]

where \( \mu \in \mathbb{R} \) and \( \kappa \in \mathbb{R}^+ \) denote the mean angle and concentration parameter, respectively. The mixture of vMDs [10] is a useful descriptor that depends on the angle between sample points; the mixture of vMDs around \( p \in S \) is defined by

\[
v(\theta; p, \kappa) = \frac{1}{|S| - 1} \sum_{q \in S \setminus \{p\}} f(\theta; \text{ang}(q - p), \kappa), \tag{24}
\]

where \( \text{ang}(q - p) \) is the angle of deviation, determined by the angle between the x-axis and the vector \( q - p \).

For example, consider the sample set \( \hat{S} = \{(0, 0), (0, 1), (1, 1), (3, 1), (2, 2)\} \). If the origin is the focusing point \( p \in \hat{S} \), then its mixture of vMDs, \( v(\theta; p, 30) \), is as plotted in Fig. 4.

For all \( p \in S_1 \) and \( q \in S_2 \), the MCF for the mixture of vMDs is the symmetric divergence given by

\[
C_{vMDs}(p, q) = \frac{1}{2} D(v(\cdot; p, \kappa) \| v(\cdot; q, \kappa)) + \frac{1}{2} D(v(\cdot; q, \kappa) \| v(\cdot; p, \kappa)), \tag{25}
\]

where \( D \) denotes the information divergence [21] between the probability densities. As for parameter setting, we need to set only the concentration parameter \( \kappa \).

The mixture of conditional Johnson–Wehrly distributions (CJWDs) [22] is an alternative version of angle-based descriptors; the mixture of CJWDs around \( p \in \hat{S} \) is defined by

\[
u(\theta; p, \kappa) = \frac{1}{|S| - 1} \sum_{q \in \hat{S} \setminus \{p\}} f(\theta; \text{ang}(q - p), \kappa, \xi(p, q)), \tag{26}
\]

where \( \kappa \) is the concentration parameter, which is the same as that in (23), and \( \xi(p, q) = ||p - q|| / \xi(p) \), where \( \xi(p) \) is the mean distance around \( p \), say,

\[
\xi(p) = \frac{1}{|S| - 1} \sum_{q \in S \setminus \{p\}} ||p - q||. \tag{27}
\]

Because the vMD \( f \) in (26) can be rewritten as a CJWD [23], the density \( u \) is referred to as the mixture of CJWDs. The MCF for the mixture of CJWDs is the same as the symmetric divergence obtained by substituting \( u \) for \( v \) in (25). As with the mixture of vMDs, we need to set only \( \kappa \). We omit an example of the mixture of CJWDs because its density shape is similar to the mixture of vMDs illustrated in Fig. 4.

4.2 Dataset and Assessment

We tested our scheme using two datasets for shape matching. The difference between the two datasets is that shape matching with LDs was easy for the first dataset but not for
the second dataset. We show that our scheme is effective regardless of the difference.

1) Handwritten Symbol Drawing Dataset

The handwritten symbol drawing dataset, available in [24], was used in [13] for contour sampling. All the drawings were classified into five shape classes, each containing 10 drawings. Using the contour sampling method of [13], some sample points were extracted automatically from each drawing to compose the sample set. Regarding the sampling method, we set the number of sample points to 16. Because the sampling method thinned out sample points that were too close together, the mean of the resultant number of sample points was 15.1. Figure 5 shows examples of drawings. In the figure, the dots on each drawing represent the contour points, and the red dots designate the sample points. The directions of these drawings are aligned, so that rotation-dependent LDs, such as the SC, work well. Because all the sample sets were almost the same size, the second term, $\sigma$, in (2) and (12) was set to zero, for simplicity.

The matching between drawings was evaluated in terms of the retrieval rate, by the so-called bullseye test [7], [10], [11]. In this test, every drawing shape was chosen as a query and matched against all shapes in the dataset. There were ten shapes in each class of symbol, and 50 shapes in total. The number of correct matches in the top ten matches was counted by obtaining the shapes found in the first ten most similar matches and checking them for the dissimilarity with each LD. The maximum number of correct matches was $10 \times 50$, when all the shapes were chosen as queries. Thus, the overall retrieval rate was calculated by dividing the number of correct matches by $10 \times 50$.

2) Leaf Image Dataset

We tested using another dataset, which contains 168 Quercus leaf images collected on the campus of the Georgia Institute of Technology. These images can be downloaded from the webpage of the cleared leaf image database [25]. The leaf shapes are classified into ten classes according to the species of Quercus. Some leaves are similar in shape even if they are in different classes. In addition, they have several torn holes, dark stains, or folds on their leaf bodies. As a result, they pose a challenge to shape matching with LDs. We extracted leaf contours using a thinning operation for the binarized leaf images. In the same manner as contour sampling, in the handwritten symbol drawing dataset, some sample points were extracted automatically from each leaf contour to compose the sample set. Figure 6 shows these examples. Again, the directions of these leaf images were aligned, and because all the sample sets were almost the same size, the second term, $\sigma$, in (2) and (12) was set to zero.
According to the bullseye test, every shape of leaf image was chosen as a query and matched against all shapes in the dataset. We counted the top $m$ matches using each LD, where $m$ denotes the size of the corresponding class. For example, regarding the “acutissima” class, we counted the top 18 matches because there were 18 acutissima leaves in the dataset. The maximum number of correct matches was 2888, when all the shapes were used in turn as queries. Consequently, we obtained the overall retrieval rate for these top matches by dividing the number of correct matches by 2888.

4.3 Results

Tables 1 and 2 show the retrieval rates and execution times for the conventional scheme, for determining $M$, and our scheme, for determining $M^*$, using the four LDs on each dataset. The execution time is the average time of ten trials on a personal computer with a 4-GHz Intel Core-i7 CPU, DDR3 32 GB, and Windows 10. The programs used to obtain these results were written in R [26]. The parameter settings of the LDs are summarized in Table 3. These parameters were carefully tuned for each dataset, to ensure that the LDs worked well. The tolerance $\epsilon$ of our scheme was set to 0.1, 0.06, 0.34, and 0.2 in the DS, SC, mixture of vMDs, and mixture of CJWDs, respectively, for the symbol drawing dataset. It was set to 0.12, 0.06, 0.29, and 0.17 in the DS, SC, mixture of vMDs, and mixture of CJWDs, respectively, for the leaf image dataset.

We observe from Tables 1 and 2 that, although there were increases in execution time when we used our scheme, the retrieval rates were improved somewhat, regardless of whether the retrieval rate with the conventional scheme was high. Among the four LDs, the mixture of vMDs was the most effective for our scheme. For execution time, when the LD was the mixture of CJWDs, for example, there was roughly a 0.9-second increase† per shape for the symbol drawing dataset, and a 2.8-second increase‡‡ per shape for the leaf image dataset, compared with the conventional scheme. Thus, the BB algorithm successfully eliminated candidate solutions that were not optimal. Consequently, because our scheme can use the LDs without modification, our scheme is worth using for shape matching, even though its computational cost is slightly higher than that of the conventional scheme.

4.4 Discussion

Because space is limited, we present two representative examples in which the correspondence is improved by our

---

Table 1. Retrieval rate (%) and execution time (s) for the handwritten symbol drawing dataset.

| LD                  | Conventional scheme | Our scheme (values of $k$ and $\alpha$) |
|---------------------|---------------------|----------------------------------------|
| DS                  | 52.8%               | 53% ($k = 7, \alpha = 0.1$)            |
|                     | 328.99 s            | 995.04 s (13.3 s increase per shape)   |
| SC                  | 92%                 | 93% ($k = 5, \alpha = 0.1$)           |
|                     | 66.83 s             | 101.27 s (0.7 s increase per shape)    |
| Mixture of vMDs     | 99.6%               | 99.8% ($k = 5, \alpha = 0.3$)         |
|                     | 82.05 s             | 118.75 s (0.7 s increase per shape)    |
| Mixture of CJWDs    | 98.6%               | 98.8% ($k = 4, \alpha = 0.2$)         |
|                     | 85.31 s             | 128.35 s (0.9 s increase per shape)    |

---

Table 2. Retrieval rate (%) and execution time (s) for the leaf image dataset.

| LD                  | Conventional scheme | Our scheme (values of $k$ and $\alpha$) |
|---------------------|---------------------|----------------------------------------|
| DS                  | 28.84%              | 28.88% ($k = 1, \alpha = 0.1$)         |
|                     | 4515.32 s           | 12067.15 s (45.0 s increase per shape)  |
| SC                  | 42.80%              | 42.87% ($k = 5, \alpha = 0.1$)         |
|                     | 968.94 s            | 1421.43 s (2.7 s increase per shape)    |
| Mixture of vMDs     | 43.56%              | 44.29% ($k = 9, \alpha = 0.1$)         |
|                     | 981.00 s            | 1385.64 s (2.4 s increase per shape)    |
| Mixture of CJWDs    | 42.14%              | 42.28% ($k = 8, \alpha = 0.3$)         |
|                     | 974.83 s            | 1444.47 s (2.8 s increase per shape)    |

---

Table 3. Parameter settings of LDs for each dataset.

| LD      | Symbol drawing | Leaf image |
|---------|----------------|------------|
| DS      | $n = 5$        | $n = 5$    |
| SC      | $\psi = 3$     | $\psi = 3$ |
|         | $\Psi = [80, 160, 320]$ | $\Psi = [90, 180, 360]$ |
|         | $\eta = 12$    | $\eta = 12$ |
| Mixture of vMDs | $\kappa = 30$ | $\kappa = 30$ |
| Mixture of CJWDs | $\kappa = 30$ | $\kappa = 30$ |

---

scheme, using the mixture of vMDs. In Figs. 7 and 8, the red dot and blue line represent a sample point and a correspondence between sample points, respectively. The green line depicts a different correspondence between the two schemes. By checking these correspondences, we can confirm that our scheme corrected the wrong correspondence yielded by the conventional scheme. In Fig. 7, we observe that our scheme corrected the correspondences on the face part of a wheelchair symbol, and Fig. 8 shows that our scheme corrected the correspondence on the small leaflet apexes. Such a visual check of matching allows human ex-
experts to understand the reason why a shape was retrieved from the dataset. This is a benefit of explainable artificial intelligence.

5. Conclusion

Using the density ratio based on the two types of probability density estimator, we presented an effective scheme for shape matching that can compensate for errors in LDs of various types. Our scheme is convenient for the LDs to apply because it does not require any modification to the LDs. Although our scheme can increase the chance of finding an appropriate correspondence, it increases the number of correspondences that we have to check for matching. Accordingly, we check the correspondences efficiently using the BB algorithm. As a result, in our experiments with two shape datasets, we observed that our scheme provided a better correspondence than the conventional scheme, even though our scheme required only a small increase in execution time.

As discussed in Sect. 3.5, at present, we do not conduct automatic parameter setting for our scheme. Introducing an easier method of setting parameters is an important topic for future study. Future verification of our scheme using a more complicated dataset is also an essential future task. However, it must be noted that for this verification, we need an effective LD for such a complicated dataset because our scheme does not work well if the LD is utterly useless. Another prospective research topic is to develop an optimization scheme to solve the minimization problem in (10). In fact, this problem is a type of set partitioning problem (see [27], [28] for background material) on the sample set $\tilde{S}_1$. Hence, applying some advanced knowledge about the set partitioning problem to this minimization problem would be interesting.

Acknowledgments

We thank Professor Tetsuyuki Takahama of Hiroshima City University for his comments on an earlier version of this paper. This work was supported in part by KAKENHI-KAKUTOKU-SHIENHI of Hiroshima City University. We thank Edanz Group (https://en-author-services.edanzgroup.com/) for editing a draft of this manuscript.

References

[1] C.R. Qi, L. Yi, H. Su, and L.J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” Advances in Neural Information Processing Systems 30, ed. I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, pp.5099–5108, Curran Associates, Inc., 2017.
[2] J. Li, B.M. Chen, and G.H. Lee, “SO-Net: Self-organizing network for point cloud analysis,” Proc. 2018 IEEE/CVF Conference on Comput. Vis. Pattern Recognit., Salt Lake City, UT, pp.9397–9406, IEEE Computer Society, IEEE, June 2018.
[3] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, and J.M. Solomon, “Dynamic graph CNN for learning on point clouds,” ACM Trans. Graphics, vol.38, no.5, Article No. 146, Oct. 2019.
[4] D.W. Aha, T. Darrell, M. Pazzani, D. Reid, C. Sammut, and P. Stone, “IJCAI 2017 workshop on explainable artificial intelligence,” http://home.earthlink.net/~dwaha/research/meetings/ijcai17-xai/, 2017.
[5] F.K. Došilović, M. Brčić, and N. Hlupić, “Explainable artificial intelligence: A survey,” Proc. 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, Opatija, Croatia, pp.210–215, Croatian Society for Information and Communication Technology, Electronics and Microelectronics, IEEE, May 2018.
[6] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (XAI),” IEEE Access, vol.6, pp.52138–52160, Sept. 2018.
[7] C. Grigorescu and N. Petkov, “Distance sets for shape filters and shape recognition,” IEEE Trans. Image Process., vol.12, no.10, pp.1274–1286, Oct. 2003.
[8] S. Manay, D. Cremers, B.W. Hong, A.J. Yezzi Jr., and S. Soatto, “Integral invariants for shape matching,” IEEE Trans. Pattern Anal. Mach. Intell., vol.28, no.10, pp.1602–1618, Oct. 2006.
[9] S. Belongie, G. Mori, and J. Malik, “Matching with shape contexts,” in Statistics and Analysis of Shapes, ed. H. Krim and A. Yezzi Jr., Modeling and Simulation in Science, Engineering and Technology, pp.81–105, Birkhäuser, Boston, 2006.
[10] K. Ueoaki, K. Iwata, N. Suematsu, and A. Hayashi, “Matching handwritten line drawings with von Mises distributions,” IEICE Trans. Inf. & Syst., vol.E104-D, no.12, pp.2487–2494, Dec. 2011.
[11] H. Fan, Y. Cong, and Y. Tang, “Self-closed partial shape descriptor for shape retrieval,” Proc. 19th IEEE Int. Conf. Image Process., pp.505–508, IEEE Signal Process. Society, Orlando, FL, Sept. 2012.
[12] K. Iwata, “Placing landmarks suitably for shape analysis by optimization,” Proc. 21st Int. Conf. Pattern Recognit., pp.2359–2362, IAPR, IEEE, Tsukuba, Japan, Nov. 2012.
[13] K. Ose, K. Iwata, and N. Suematsu, “Sampling shape contours using optimization over a geometric graph,” IJCAI Trans. Inf. & Syst., vol.E102-D, no.12, pp.2547–2556, Dec. 2019.
[14] B.W. Silverman, Density estimation for statistics and data analysis, Monographs on statistics and applied probability, vol.26, Chapman & Hall/CRC Press, Boca Raton, FL, 1998.
[15] C. Bishop, Pattern Recognition and Machine Learning, Information Science and Statistics, Springer-Verlag, New York, 2006.
[16] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, 3rd ed., MIT Press, Cambridge, MA, July 2009.
[17] J. Clausen, “Branch and bound algorithms – principles and examples,” https://imada.sdu.dk/~jbj/heuristikker/TSPtext.pdf, March 1999.
[18] K. Iwata, “Reducing the computational cost of shape matching with the distance set,” Proc. 23rd Int. Conf. Pattern Recognit., Cancun,
[19] R. von Mises, “Über die ‘ganzzahligkeit’ der atomgewichte und verwandte fragen,” Physikalische Zeitschrift, vol.19, pp.490–500, 1918, also in Selecta II, 15–34.

[20] K.V. Mardia and P.E. Jupp, Directional Statistics, Wiley, Chichester, Nov. 1999.

[21] T.M. Cover and J.A. Thomas, Elements of Information Theory, 2nd ed., John Wiley & Sons, Inc., New York, 2006.

[22] K. Iwata, N. Suematsu, and A. Hayashi, “A new shape descriptor based on an angular-linear probability distribution,” Proc. 3rd IAPR Asian Conference on Pattern Recognit., pp.484–488, IAPR, IEEE, Kuala Lumpur, Malaysia, Nov. 2015.

[23] R.A. Johnson and T.E. Wehrly, “Some angular-linear distributions and related regression models,” J. American Statistical Association, vol.73, no.363, pp.602–606, Sept. 1978.

[24] K. Ose and K. Iwata, “Data set of line drawings with different stroke order and number,” http://www.prl.info.hiroshima-cu.ac.jp/kiwata/oseiwa/, May 2017.

[25] C. Price, “Cleared leaf image database: Georgia tech,” http://clearedleavesdb.org/?q=content/georgia-tech, April 2013.

[26] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2020. https://www.R-project.org/.

[27] E. Balas and M.W. Padberg, “Set partitioning: A survey,” SIAM Review, vol.18, no.4, pp.710–760, 1976.

[28] K. Hoffman and M. Padberg, “Set covering, packing and partitioning problems,” in Encyclopedia of Optimization, ed. C.A. Floudas and P.M. Pardalos, pp.3482–3486, Springer, Boston, MA, 2009.

Kazunori Iwata received BE and ME degrees from Nagoya Institute of Technology, Aichi, Japan, in 2000 and 2002, respectively, and a Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 2005. He was a JSPS research fellow from April 2002 to March 2005. He has been with Hiroshima City University, Hiroshima, Japan, since April 2005. His research interests include machine learning, pattern recognition, and information theory. He was an Associate Editor for IEICE Transactions on Information and Systems from July 2013 to June 2017.

Kazushi Mimura received his B.E., M.E., and Ph.D. (Sci.) degrees in 1992, 1994, and 1999, respectively, from Osaka University, Osaka, Japan. He is now a professor of the Graduate School of Information Sciences, Hiroshima City University, Hiroshima, Japan. In 2008, he was an academic visitor at the Disordered Systems Group, King’s College London, London, U.K. His research interests include information theory, machine learning, and statistical mechanics. He is also a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Physical Society of Japan (JPS).

Hiroki Yamamoto received BS and MS degrees from Hiroshima City University, Hiroshima, Japan, in 2017 and 2019, respectively. He was a student member of IEICE until March 2019. He joined DISCO Corporation in April 2019.