Escherization with a Distance Function Focusing on the Similarity of Local Structure

Yuichi Nagata(✉)

Tokushima University, 2-24, Shinkura-cho, Tokushima 770-8501, Japan
nagata@is.tokushima-u.ac.jp

Abstract. The Escherization problem is that, given a goal figure, find a closed figure that is as close as possible to the goal figure and tiles the plane. In the Koizumi and Sugihara's formulation for the Escherization problem, the tile and goal shapes are represented as polygons whose similarity is evaluated by the Procrustes distance. In this paper, we incorporate a new distance function into their formulation, aiming at finding more satisfiable tile shapes. The proposed distance function successfully picks up tile shapes that are intuitively similar to the goal shape even when they are somewhat different from the goal shape in terms of the Procrustes distance. Due to the high computational cost for solving the formulated problem, we develop a tabu search algorithm to tackle this problem.

Keywords: Escher tiling · Tiling · Similarity measure

1 Introduction

A tiling refers to any pattern that covers the plane without any gaps or overlap. The Dutch artist M. C. Escher is famous for creating many artistic tilings, each of which consists of a few recognizable (especially one) figures such as animals. Such tiling is now called Escher tiling and it is a very intellectual task to design artistic Escher tilings while satisfying the constraints imposed to realize tiling.

As an attempt to automatically generate Escher tilings, Kaplan and Salesin [5] introduced the following optimization problem. Given a closed plane figure S (goal figure), find a closed figure T such that (i) T is as close as possible to S, and (ii) copies of T fit together to form a tiling of the plane. This problem is called the Escherization problem named after Escher. Koizumi and Sugihara [6] showed that when both tile and goal shapes are represented as polygons, the Escherization problem can be formulated as an eigenvalue problem.

Several enhancements to the Koizumi and Sugihara's formulation have been proposed. Imahori and Sakai [3] parameterized tile shapes (polygons) in a more flexible way, which creates a great deal of flexibility in the possible tile shapes (extended Koizumi and Sugihara's formulation). It requires, however, a considerable computational effort to solve the Escherization problem formulated with

A. Auger et al. (Eds.): PPSN 2018, LNCS 11101, pp. 108–120, 2018.
https://doi.org/10.1007/978-3-319-99253-2_9

this extension and they developed a local search algorithm for this problem. In the original Koizumi and Sugihara's formulation, the Procrustes distance [7] was introduced to measure the similarity between the tile and goal shapes. Imahori et al. [4], however, suggested that the Procrustes distance does not necessarily reflect an *intuitive* similarity between the two shapes. To handle this issue, they introduced weights to the Procrustes distance to emphasize the similarity with important parts of the goal figure. The idea of the weighted Procrustes distance, however, has not been incorporated into the extended Koizumi and Sugihara's formulation due to the heavy computational cost of calculating the weighted Procrustes distance.

In this paper, we propose another similarity measure (distance function), which captures the similarity of local structures between the tile and goal shapes, to successfully evaluate an intuitive similarity between them. We incorporate this similarity measure into the extended Koizumi and Sugihara's formulation and apply a tabu search algorithm [9] to the Escherization problem obtained.

2 Related Work

We first explain basic knowledge of tiling and then explain the Koizumi and Sugihara's formulation of the Escherization problem along with extended studies.

2.1 Isohedral Tilings

A *monohedral* tiling is one in which all the tiles are the same shape. If a monohedral tiling has a repeating structure, this tiling is called *isohedral*. There are 93 different types of isohedral tilings [1], which are individually referred to as IH1, IH2, ..., IH93. Figure 1 illustrates an example of an isohedral tiling belonging to IH47 with a few technical terms. A *tiling vertex* is a point where at least three tiles meet. A *tiling edge* is a boundary surface where exactly two tiles meet. A tiling polygon is the polygon formed by connecting the tiling vertices of a tile.

For each IH type, the nature of tile shapes can be represented by a template [8]. A template represents a tiling polygon from which all possible tile shapes are obtained by deforming the tiling edges and moving the tiling vertices under the constraints specified by the template. For example, Fig. 1 illustrates a template of IH47; this template means that the tiling polygon is a quadrilateral consisting of two opposite J edges that are parallel to one another and two independent S edges. There are four types of tiling edges (types J, S, U, and I) and it is convenient to express these types with colored arrowheads as illustrated in Fig. 1 (only types J and S are shown). These types are closely related to how the tiles are fitted to each other, and a template also gives information about the adjacency relationship between the tiles.

According to the adjacency relationship, four types of tiling edges can be deformed in the following ways (see also Fig. 1). A type J edge can be deformed in any arbitrary fashion, but the corresponding J edge must also be deformed into the same shape. A type S edge must be symmetric with respect to the

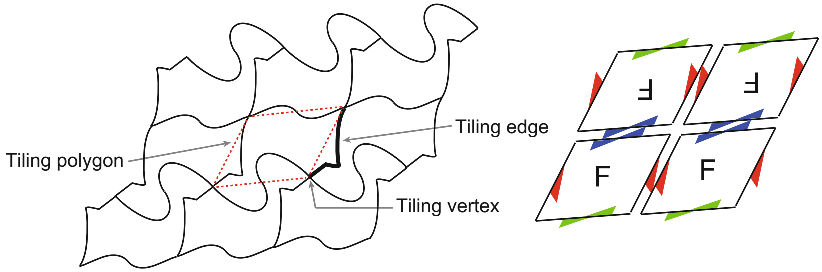

Fig. 1. Example of an isohedral tiling (left), and the template of IH47 (right) where J and S edges are indicated by single arrowheads and facing arrowheads, respectively.

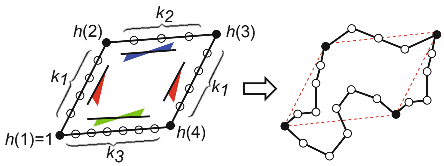

Fig. 2. Template of IH47 for a specific assignment of the points to the tiling edges (left), and an example of a tile shape (right).

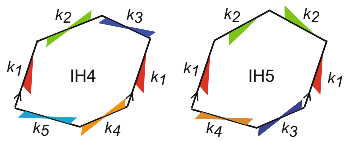

Fig. 3. Templates of IH4 and IH5. Two opposite J edges marked with $\wedge$ are parallel to one another.

midpoint. A type U edge must be symmetric with respect to a line through the midpoint and orthogonal to it. A type I edge must be a straight line.

2.2 Koizumi and Sugiharas's Formulation and Its Extension

Koizumi and Sugihara [6] modeled the tile shape as a polygon of n points. In this case, the template of IH47 is represented as shown in Fig. 2, where exactly one point must be placed at each of the tiling vertices (black circles) and the remaining points are placed on the tiling edges (white circles). This template represents possible arrangements of the n points; the n points can be moved as illustrated in Fig. 2. Koizumi and Sugihara originally placed the same number of points on every tiling edge. After that, Imahori et al. [3] extended this model to assign different numbers of points on the tiling edges (extended Koizumi and Sugihara's formulation). We denote the numbers of points placed on the tiling edges as $k_1, k_2, \cdots$ as illustrated in Fig. 2.

Let the n points on the template be indexed clockwise by $1, 2, \ldots, n$, starting from one of the tiling vertices. We represent the tile shape as a $2n$-dimensional vector $\boldsymbol{u} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n)^\top$, where $(x_i, y_i)^\top$ is the coordinates of the ith point in the xy-plane. We will also denote $(x_i, y_i)^\top$ as $\hat{\boldsymbol{u}}_i$. We refer to the tile shape (polygon) specified by $\boldsymbol{u}$ as U. The values of vector $\boldsymbol{u}$ are constrained so that the tile shape U is consistent with the template selected. For example, if we select IH47, the values of vector $\boldsymbol{u}$ must satisfy the following equation:

$$\begin{cases} \hat{\boldsymbol{u}}_{h(1)+i} - \hat{\boldsymbol{u}}_{h(1)} = \hat{\boldsymbol{u}}_{h(4)-i} - \hat{\boldsymbol{u}}_{h(4)} & (i = 1, \ldots, k_1 + 1) \\ \hat{\boldsymbol{u}}_{h(2)+i} - \hat{\boldsymbol{u}}_{h(2)} = -(\hat{\boldsymbol{u}}_{h(3)-i} - \hat{\boldsymbol{u}}_{h(3)}) & (i = 1, \ldots, \lfloor \frac{k_2+1}{2} \rfloor) \\ \hat{\boldsymbol{u}}_{h(4)+i} - \hat{\boldsymbol{u}}_{h(4)} = -(\hat{\boldsymbol{u}}_{n+1-i} - \hat{\boldsymbol{u}}_{h(1)}) & (i = 1, \ldots, \lfloor \frac{k_3+1}{2} \rfloor) \end{cases}, \tag{1}$$

where $h(s)$ $(s = 1, \ldots, 4)$ is the index of the sth tiling vertex as shown in Fig. 2.

Equation (1) is a homogeneous system of linear equations and is represented by

$$A\boldsymbol{u} = \boldsymbol{0}, \tag{2}$$

where A is a $m' \times 2n$ matrix $(m' < 2n)$. Let $\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_m}$ be the orthonormal basis of $\mathrm{Ker}(A)$. A general solution of Eq. (2) is then given by

$$\boldsymbol{u} = \xi_1 \boldsymbol{b_1} + \xi_2 \boldsymbol{b_2} + \cdots + \xi_m \boldsymbol{b_m} = B\boldsymbol{\xi}, \tag{3}$$

where $B = (\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_m})$ is a $2n \times m$ matrix and $\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_m)^\top$ is a parameter vector. In fact, tile shapes for every isohedral tilings can be parameterized in the form of Eq. (3), where the matrix B depends on the assignment of the n points to the tiling edges as well as isohedral type.

In the Koizumi and Sugihara's formulation, the goal figure is also represented as a polygon of n points and their coordinates are represented by a $2n$-dimensional vector $\boldsymbol{w} = (x_1^w, x_2^w, \ldots, x_n^w, y_1^w, y_2^w, \ldots, y_n^w)^\top$, where $(x_i^w, y_i^w)^\top$ is the coordinates of the ith point of the goal polygon. We will also denote $(x_i^w, y_i^w)^\top$ as $\hat{\boldsymbol{w}}_i$. We refer to the goal shape (polygon) specified by $\boldsymbol{w}$ as W. To measure the similarity between the two polygons U and W, they employed the Procrustes distance [7]. Let us first, however, explain a more simple but essentially the same distance measure for the ease of understanding. We refer to this distance measure as the normal distance in this paper. The square of the normal distance between the two polygons U and W is defined by

$$d^2(U, W) = \|\boldsymbol{u} - \boldsymbol{w}\|^2 = \sum_{i=1}^{n} \|\hat{\boldsymbol{u}}_{\boldsymbol{i}} - \hat{\boldsymbol{w}}_{\boldsymbol{i}}\|^2, \tag{4}$$

where $\|\cdot\|$ is the Euclidean norm.

When the normal distance is used, from Eqs. (3) and (4), the Escherization problem can be formulated as the following unconstrained optimization problem:

$$\text{minimize: } \|B\boldsymbol{\xi} - \boldsymbol{w}\|^2. \tag{5}$$

This is a least-squares problem and the solution is given by $\boldsymbol{\xi}^* = (B^\top B)^{-1} B^\top \boldsymbol{w} = B^\top \boldsymbol{w}$ with the minimum value $-\boldsymbol{\xi}^{*\top} \boldsymbol{\xi}^* + \boldsymbol{w}^\top \boldsymbol{w}$. The optimal tile shape $\boldsymbol{u}^*$ is then obtained by $\boldsymbol{u}^* = B\boldsymbol{\xi}^*$.

When calculating the normal distance between the two polygons, we need to consider the n different numbering for the goal polygon W by shifting the first point for the numbering. Therefore, we define $\boldsymbol{w_j}$ $(j = 1, 2, \ldots, n)$ in the same way as $\boldsymbol{w}$ by renumbering the index of the n points such that the jth point (in the original index) becomes the first point.

Let I be a set of the indices for the isohedral types and K_i a set of all possible configurations for the assignment of the n points to the tiling edges for an isohedral type i. For example, $K_{47} = \{(k_1, k_2) \mid 0 \le k_1, 0 \le k_2, 2k_1 + k_2 \le n-4\}$ whereas k_3 is determined by $k_3 = n - 4 - (2k_1 + k_2)$ (see Fig. 2). Because the matrix B depends on $i \in I$ and $k \in K_i$, we denote it as B_{ik}. Let $J = \{1, 2, \ldots, n\}$ be a set of the indices of the first point for the n different numbering of the goal polygon. If we try to perform the exhaustive search, we need to compute

$$\min_{\boldsymbol{\xi} \in R^m} \|B_{ik}\boldsymbol{\xi} - \boldsymbol{w}_j\|^2 = -{\boldsymbol{\xi}^*_{ikj}}^\top \boldsymbol{\xi}^*_{ikj} + \boldsymbol{w}^\top \boldsymbol{w}, \tag{6}$$

for all combinations of $i \in I$, $k \in K_i$, and $j \in J$, where $\boldsymbol{\xi}^*_{ikj} = B_{ik}^\top \boldsymbol{w}_j$.

For each $i \in I$ and $k \in K_i$, it takes $O(n^3)$ time to compute Eq. (6) for all values of $j \in J$ because it takes $O(n^3)$ time for computing B_{ik} and $O(n^2)$ time for computing $B_{ik}^\top \boldsymbol{w}_j$ (for each value of j). However, the order of K_i reaches $O(n^3)$ for IH5 and IH6 and $O(n^4)$ for IH4, and it requires a considerable computational time to perform the exhaustive search. Figure 3 shows the templates of IH4 and IH5. To alleviate this problem, Imahori and Sakai [4] proposed a local search algorithm to search for only promising configurations in K_i, which has succeeded in finding better tile shapes than the original Koizumi and Sugihara's method.

Finally, we mention the difference between the Procrustes distance and the normal distance. For some isohedral types including IH47 and IH4, exactly the same result is obtained with either distance measure. For some isohedral types, however, the Procrustes distance must be used because the templates can only parameterize tile shapes facing in a specific direction. For example, the two adjacent J edges in the template of IH5 (see Fig. 3) are parameterized such that they make equal and opposite angles with the y-axis. The Procrustes distance calculates the normal distance after rotating U so that the normal distance between U and W is minimized. When the Procrustes distance is used, the Escherization problem can be reduced to an eigenvalue problem, which can be solved in $O(n^2)$ time [4]. We use only the normal distance to explain the original Koizumi and Sugihara's formulation [6], the subsequent studies [2–4], and the proposed method for the ease of understanding and due to space limitations.

2.3 The Weighted Normal Distance

The normal distance Eq. (4) seems to be the most natural similarity measure between two polygons. Imahori et al. [4], however, suggested that the normal distance does not necessarily reflect intuitive similarity between two polygons. The main cause is that in many cases goal figures have important parts that characterize their shapes and they assigned *weights* to the points on the important parts of the goal polygon to emphasize the similarity with these parts.

Let k_i $(i = 1, 2, \ldots, n)$ be a positive weight assigned to the ith point of the goal polygon W. The weighted normal distance is then defined by

$$d_w^2(U, W) = \sum_{i=1}^{n} k_i \|\hat{\boldsymbol{u}}_i - \hat{\boldsymbol{w}}_i\|^2 = \boldsymbol{u}^\top G \boldsymbol{u} - 2\boldsymbol{w}^\top G \boldsymbol{u} + \boldsymbol{w}^\top G \boldsymbol{w}, \tag{7}$$

where G is the $2n \times 2n$ diagonal matrix whose diagonal elements are $k_1, k_2, \ldots, k_n,\ k_1, k_, \ldots, k_n$. When the weighted normal distance is used, from Eqs. (3) and (7), the Escherization problem is formulated as follows:

$$\text{minimize: } \boldsymbol{\xi}^\top B^\top G B \boldsymbol{\xi} - 2\boldsymbol{w}^\top G B \boldsymbol{\xi} + \boldsymbol{w}^\top G \boldsymbol{w}. \tag{8}$$

3 Proposed Method

We propose a new distance function to evaluate intuitive similarity between two polygons and incorporate it into the extended Koizumi and Sugihara's formulation. We try to solve the formulated problem using tabu search (TS) [9] to search for as many configurations $k \in K_i$ as possible for each isohedral type $i \in I$.

3.1 The Proposed Similarity Measure

As expressed by Eqs. (4) and (7), in order to shorten the (weighted) normal distance, the points of the tile polygon U must be close to the corresponding points of the goal polygon W. In contrast, we focus on the similarity of the relative positional relationship of adjacent points between two polygons. The proposed distance function is defined as follows:

$$d_a^2(U, W) = \sum_{i=1}^{n} k_i \|(\hat{\boldsymbol{u}}_{\boldsymbol{i+1}} - \hat{\boldsymbol{u}}_{\boldsymbol{i}}) - (\hat{\boldsymbol{w}}_{\boldsymbol{i+1}} - \hat{\boldsymbol{w}}_{\boldsymbol{i}})\|^2, \tag{9}$$

where $n+1$ represents 1 and k_i is the weight. We refer to the proposed distance as the (weighted) *adjacent difference (AD) distance*.

In the right side of Fig. 4, we can see a typical example of a tile polygon (red line) that is determined to be very similar to the goal polygon "bat" under the AD distance but is not so under the normal distance. The middle of the figure shows the opposite situation. Compared to the tile shape in the middle of the figure, the tile shape in the right side does not so much overlap with the goal polygon, but it seems to be intuitively more similar to the goal polygon than the former one. The reason is that local shapes of the contours of the wings and ears are well preserved in the right side figure even though overall shape is distorted (e.g., the vertical width of the wings is getting narrower). As exemplified in this example, even if the global structure is somewhat distorted, it would be better to actively preserve local structures of the goal shape to search for more satisfiable tile shapes. The AD distance is designed assuming such a situation.

It is also possible to assign weights to edges of the goal polygon W and k_i in Eq. (9) is the weight assigned to the edge between ith and $(i+1)$th points. In fact, Eq. (9) can be expressed by the same matrix representation as the right side of Eq. (7), where G is a $2n \times 2n$ symmetric tridiagonal matrix whose non-zero elements are defined as follows:

$$\begin{cases} g_{i,i} &= g_{i+n,i+n} &= k_i + k_{i+1} \\ g_{i,i+1} &= g_{i+n,i+1+n} &= -k_i \\ g_{i+1,i} &= g_{i+1+n,i+n} &= -k_i \end{cases}, \tag{10}$$

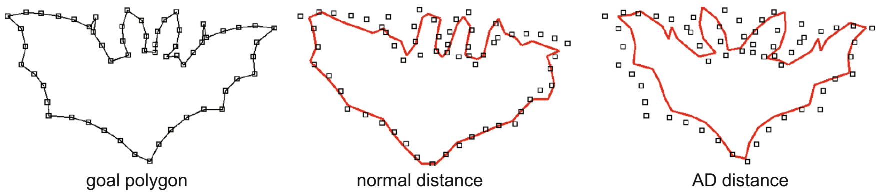

Fig. 4. Goal polygon "bat" and tile shapes that are very close to the goal polygon under the normal distance and the AD distance, respectively.

where $2n + 1$ means 1. We should note that the matrix G depends on the first point for the numbering of the goal polygon j ($\in J$) (this also applies to the case where the weighted normal distance is used) because the indices of the weighted edges must be also shifted depending on the numbering. Therefore, we define G_j ($j = 1, 2, \ldots, n$) in the same way as Eq. (10), by renumbering the index such that the jth point (in the original index) becomes the first point.

3.2 The Extended Koizumi and Sugihara's Formulation with the AD Distance

The weighted normal distance Eq. (7) and the AD distance Eq. (9) have the same matrix representation and the Escherization problem using the AD distance is also formulated by Eq. (8). When the AD distance is incorporated into the extended Koizumi and Sugihara's formulation, we need to solve the following optimization problem:

$$\text{minimize: } \boldsymbol{\xi}^\top B_{ik}^\top G_j B_{ik} \boldsymbol{\xi} - 2\boldsymbol{w}_j{}^\top G_j B_{ik} \boldsymbol{\xi} + \boldsymbol{w}_j{}^\top G_j \boldsymbol{w}_j, \tag{11}$$

for all combinations of $i \in I$, $k \in K_i$, and $j \in J$.

Let us consider Eq. (8) again instead of Eq. (11) for simplicity (indices i, k, and j are omitted). The solution $\boldsymbol{\xi}^*$ to Eq. (8) is obtained by solving the equation $B^\top G B \boldsymbol{\xi} = B^\top G \boldsymbol{w}$ (the minimum is $-\boldsymbol{\xi}^{*\top} \boldsymbol{\xi}^* + \boldsymbol{w}^\top \boldsymbol{w}$). However, as explained later, the matrix $B^\top GB$ is rank deficient (when the AD distance is used) and we find the solution in the following way, which is essentially the same as in [4]. First, a set of column vectors $\boldsymbol{b_1}, \boldsymbol{b_2}, \ldots, \boldsymbol{b_m}$ (see Eq. (3)) are linearly transformed into $\boldsymbol{b'_1}, \boldsymbol{b'_2}, \ldots, \boldsymbol{b'_{m'}}$ such that $\boldsymbol{b'_i}^\top G \boldsymbol{b'_j} = \delta_{ij}$ (the Kronecker delta function) for $i, j \in \{1, 2, \ldots, m'\}$ (as explained later, $m' = m-2$). Such a set of column vectors can be obtained in $O(n^3)$ time by using the Gram-Schmidt orthogonalization process with an inner product defined as $<\boldsymbol{x}, \boldsymbol{y}> = \boldsymbol{x}^\top G \boldsymbol{y}$. Let a matrix B' be defined as $B' = (\boldsymbol{b'_1}, \boldsymbol{b'_2}, \ldots, \boldsymbol{b'_{m'}})$ and the tile shape U be parameterized by $\boldsymbol{u} = B'\boldsymbol{\xi}$. Because $B'^\top GB'$ becomes an identity matrix, the solution to Eq. (8) (B is replaced with B' in this case) is obtained by $\boldsymbol{\xi}^* = B'^\top G\boldsymbol{w}$. Note that when the AD distance is used, $m' = m-2$ because the matrix G is rank deficient by 2. Therefore, the degree of freedom for parameterizing tile shapes is also reduced. Intuitively, this is because the value of the AD distance does not depends on the position of the center of gravity of U (it does not determined uniquely).

The matrix B' depends on i ($\in I$) and k ($\in K_i$). In addition, unlike in the case of the matrix B, B' depends on j ($\in J$). Therefore, we denote the matrix B' as B'_{ikj} when specifying the indices i, k, and j. For each $i \in I$ and $k \in K_i$, it takes $O(n^4)$ time for solving the optimization problem Eq. (11) (i.e., computing $\boldsymbol{\xi}^*_{\boldsymbol{ikj}} = {B'_{ikj}}^\top G_j \boldsymbol{w_j}$) for all values of j because it takes $O(n^3)$ time for computing B'_{ikj}. Note that if the weight is not introduced, B'_{ikj} does not depend on j and it takes $O(n^3)$ time in the same situation. Remember that when the normal distance is used, this computational cost is $O(n^3)$ (see Sect. 2.2). Therefore, it is computationally more difficult to search for many configurations $k \in K_i$ for each isohedral type $i \in I$, compared to the case of the normal distance. This also applies to the case where the weighted normal distance is used, and therefore the weighted normal distance was not incorporated into the extended Koizumi and Sugihara's formulation.

3.3 A Tabu Search Algorithm

It requires a considerable computation time to solve the optimization problem Eq. (11) for all possible combinations of $i \in I$, $k \in K_i$, and $j \in J$ and we propose a TS algorithm to search for only promising configurations among them. The basic idea is similar to the local search algorithm [4] developed for the extended Koizumi and Sugihara's formulation with the normal distance (Eq. (6)), but we propose a TS algorithm here to enhance the performance. Compared to the case of the normal distance, however, the required computational cost is significantly increased and we need to reduce the computational cost.

The TS algorithm is performed for each isohedral type $i \in I$. In the TS algorithm, the solution candidate represents a configuration (k, j) and its objective value is given by $-{\boldsymbol{\xi}^*_{\boldsymbol{ikj}}}^\top \boldsymbol{\xi}^*_{\boldsymbol{ikj}} + \boldsymbol{w}^\top \boldsymbol{w}$ (the minimum of Eq. (11)). We define the neighborhood as a set of configurations (k', j') given by the combinations of $j' \in \{j, j \pm 1\}$ and k' that is obtained by incrementing (or decrementing) the number of points assigned to two tiling edges in all possible ways. For example, if we select IH47 (see Fig. 2), for the current $k = (k_1, k_2)$, possible values of k' are $(k_1 \pm 1, k_2 \mp 2, k_3), (k_1, k_2 \pm 1, k_3 \mp 1), (k_1 \pm 1, k_2, k_3 \mp 2)$ (actually k_3 is omitted because k_3 is obtained as $n - 4 - 2k_1 - k_2$).

Algorithm 1 depicts the TS algorithm. We denote the current solution (k, j) and its neighborhood as x and $\mathcal{N}(x)$, respectively. Before starting the iterations, the current solution x and the current best solution x_{best} are initialized with a randomly generated solution (line 1). At each iteration, the best non-tabu solution x' (we define tabu solutions later) is selected from the neighborhood $\mathcal{N}(x)$ (line 3). In addition, the aspiration criterion is considered, where a solution that improves the current best solution x_{best} is always regarded as a non-tabu solution as an exception. The current solution x and current best solution x_{best} (if necessary) are then updated by x' (line 4). Iterations are repeated until the number of iterations reaches a given maximum number $iterMax$ (lines 2 and 5). Finally, the current best solution x_{best} is returned (line 7).

We explain how to define tabu solutions with an example where k is represented as (k_1, k_2, k_3, k_4). Let the current solution be denoted as $(k_1, k_2, k_3, k_4; j)$.

If the current solution is replaced with the selected solution $(k_1', k_2, k_3', k_4; j')$, two pairs of (k_1', j') and (k_3', j') are stored in the tabu list during the subsequent T iterations. At each iteration, when a neighbor solution $(k_1, k_2', k_3', k_4; j')$ is obtained from the current solution $(k_1, k_2, k_3, k_4; j)$, this solution is regarded as a tabu-solution if both (k_2', j') and (k_3', j') exists in the tabu list.

Algorithm 1. Tabu-Search(isohedral type i)

```
Set the solution x randomly, set x_best := x, and iter_no := 0;
while (iter_no ≤ iterMax) do
    Select a best non-tabu solution x' ∈ N(x) (aspiration criterion is considered);
    Update x := x' and x_best := x' (if x' is better than x_best);
    Set iter_no := iter_no + 1;
end while
return x_best;
```

We mention the main difference between the proposed TS and the local search used in [4]. In [4], the solution candidate was represented by only k and it was evaluated by testing the n different values for j. Its computational effort is $O(n^3)$ when using the normal distance. On the other hand, however, it takes $O(n^4)$ time in the same situation when the weighted AD distance is used because it takes $O(n^3)$ time for each value of j. In our observation, the optimal value of j for the value of k tends to continuously change with the change of k. Therefore, it is reasonable to restrict the search range of j around the current value of j in the neighborhood as in the proposed TS algorithm. Therefore, we have decided to include the variable j in the solution x.

Although there are 93 isohedral types, it is enough to consider only 10 isohedral types (IH1, IH2, IH3, IH4, IH5, IH6, IH7, IH8, IH21, IH28) for the optimization because the remaining 83 types are approximately obtained by assigning no point to tiling edges in the 10 isohedral types. In our observation, good tile shapes are mostly obtained with IH4, IH5, and IH6 because other isohedral types do not have enough flexibility to represent tile shapes (e.g., most tiling edges have the same shape). We therefore consider only the three isohedral types IH4, IH5, IH6 for the optimization. In fact, the order of K_i is equal to or greater than $O(n^3)$ only for these isohedral types.

Through preliminary experiments, the parameters of the TS algorithm were determined as follows: $T = 50$ and $iterMax = 100$, where we set the value of $iterMax$ to a small value since it was better to repeat the TS algorithm from different initial solutions rather than to continue one trial for a long time. We define one set of trials as 60 runs of the TS algorithm during which the top 20 tile shapes (including non-local minima) found are stored.

4 Experimental Results

We implemented the proposed TS algorithm in C++ and executed the program code on a Ubuntu 14.04 Linux PC with Intel Core i7-4790@3.60 GHz CPU.

We applied one set of trials of the TS algorithm to the three goal figures hippocampus ($n = 59$), bird ($n = 60$), and spider ($n = 126$) using each of the four distance functions (normal distance, weighted normal distance, AD distance, and weighted AD distance). The execution time of one set of TS algorithm were about 50s (normal and AD distances) and 140s (weighted normal and weighted AD distances) for the goal polygon bird and about 400s (normal and AD distances) for the goal polygon spider.

Figure 5 shows the three goal polygons followed by the intuitively best tile shapes obtained with the four distance functions. Note that the top tile shape in terms of the distance value do not necessarily the best one form an intuitive point of view, and we selected the intuitively best one among the top 20 tile shapes for each distance function, where the numbers in parentheses indicate the ranking in terms of distance values. The tile shapes are drawn with red lines on the points of the goal polygons (black points). When the (weighted) AD distance is used, the position of the center of gravity of the tile shape is not determined (see Sect. 3.2) and we put the tile such that the normal distance is minimized. When the weighted normal distance and the weighted AD distance are used, the weighted values were all set to four and the weighted points or the both ends of the weighted edges are drawn in green. In addition, Fig. 6 presents three tilings generated from the tile shapes shown on the right side of Fig. 5.

We first discuss the results for hippocampus. From the definition, when the normal distance is used, the resulting tile shape seems to be most overlapped with the goal polygon, but differences in some local structures are conspicuous. By using the weighted normal distance, the difference in the weighted part (head) is getting smaller. When using the AD distance, although the obtained tile shape does not so much overlap the goal polygon (compared to the normal distance case), local structures of the goal polygon are well maintained. The obtained tile shape seems to be intuitively quite similar to the goal polygon, except for the problem that the width of the head part is shortened and the neck is too thin. By introducing weights to the AD distance, the local shape of the head part is getting similar to that of the goal polygon and the aforementioned problem in the AD distance is somewhat improved.

Next, we discuss the results for bird. The tile shape obtained with the normal distance seems to be most overlapped with the goal polygon, but the difference in the foot part is conspicuous. Therefore, we assigned weights to the foot part as well as the beak part for the weighted normal distance. However, no particular improvement is found when the weighted normal distance is used. In contrast to the (weighted) normal distance, when using the AD distance, not only the overall structure but also the local structures (especially the foot part) are well maintained. By introducing weights to the AD distance, only the local shape of the beak parts is slightly improved.

Next, we discuss the results for spider, which is a pretty challenging goal figure. Since it was difficult to determine appropriate weight points and edges, only the normal and AD distances were tested. We can see that the tile shape obtained with the AD distance seems to be intuitively more similar to the goal

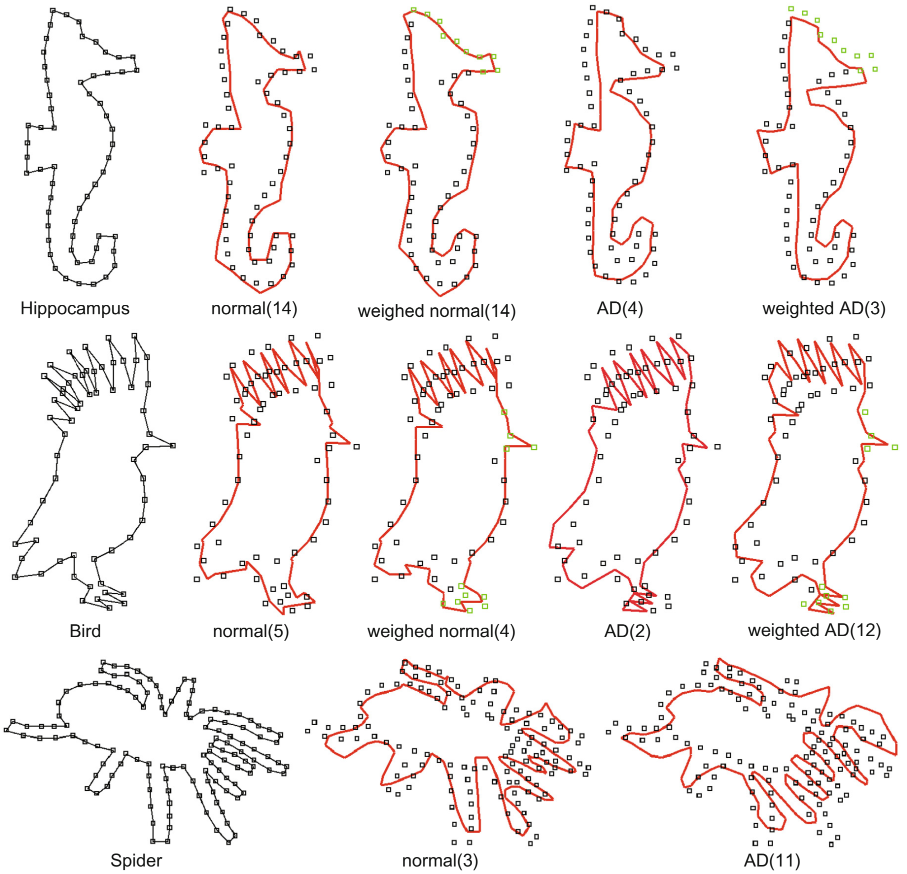

Fig. 5. The goal polygons and tile shapes obtained with the four distance functions. (Color figure online)

figure than that with the normal distance. The main reason is that the shape of each leg is well preserved, although the positions of the bases of the legs are different from those of the goal figure.

We also mention the problem of the AD distance. Compared to the normal distance, tile shapes obtained with the AD distance are rich in variety and many undesirable tile shapes are also included in the top 20 tile shapes. The reason for this is that the value of the AD distance may be small even if overall tile shape is fairly distorted from the goal shape. In the present situation, a satisfiable tile shape is obtained when the global structure happens to be similar to that of the goal figure to some extent. In such a case, we can find a very satisfiable tile shape which cannot be obtained with the (weighted) normal distance.

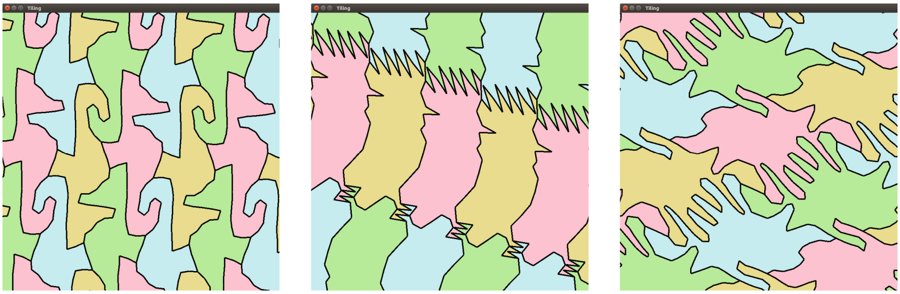

Fig. 6. Tilings generated from the tile shapes on the right side of Fig. 5. (Color figure online)

5 Conclusion

We have proposed a new distance function (AD distance), which captures the similarity of local structures between the two shapes. When the AD distance is incorporated into the (extended) Koizumi and Sugihara's formulation of the Escherization problem, tile shapes obtained actively preserve local structures of the goal shape even if the global structure is sacrificed. Experimental results showed that it is better to positively preserve local structures of the goal shape by allowing the global structure to deform (if the degree of deformation is not very large), in order to obtain intuitively satisfiable tile shapes. Due to the high computational cost of the exhaustive search for the formulated Escherization problem, we developed a TS algorithm for solving this problem, which made it possible to obtain satisfiable tile shapes in a reasonable computational time.

Acknowledgements. This work was supported by JSPS KAKENHI Grant Number 17K00342.

References

1. Grünbaum, B., Shephard, G.: Tilings and Patterns. A Series of Books in the Mathematical Sciences (1989)
2. Imahori, S., Sakai, S.: A local-search based algorithm for the Escherization problem. In: Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management, pp. 151–155 (2012)
3. Imahori, S., Sakai, S.: A local-search based algorithm for the Escher-like tiling problem. IPSJ SIG Technical reports, vol. 2013-AL-144, no.14 (2013 in Japanese)
4. Imahori, S., Kawade, S., Yamakata, Y.: Escher-like tilings with weights. In: Akiyama, J., Ito, H., Sakai, T. (eds.) JCDCGG 2015. LNCS, vol. 9943, pp. 132–142. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-48532-4_12
5. Kaplan, C.S., Salesin, D.H.: Escherization. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 499–510 (2000)
6. Koizumi, H., Sugihara, K.: Maximum eigenvalue problem for Escherization. Graph. Comb. **27**(3), 431–439 (2011)

7. Werman, M., Weinshall, D.: Similarity and affine invariant distances between 2D point sets. IEEE Trans. Pattern Anal. Mach. Intell. **17**(8), 810–814 (1995)
8. Kaplan, C.S.: Introductory Tiling Theory for Computer Graphics. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers, San Rafael (2009)
9. Glover, F., Lagunab, M.: Tabu Search. Kluwer Academic Publishers, Dordrecht (1997)