

2 種類の図形によるタイリング生成における図形の選択方法

名古屋大学大学院・工学研究科 川出 静 (Shizuka Kawade)

今堀 慎治 (Shinji Imahori)

Graduate School of Engineering,

Nagoya University

概要

有限種類の図形によって、平面を隙間も重なりもなく敷き詰めたものをタイリングと呼ぶ。また、入力図形が与えられ、その図形にできるだけ近い図形によるタイリングを求める問題をエッシャー風タイリング問題と呼ぶ。1 種類の図形に対するエッシャー風タイリング問題については近年盛んに研究されてきた。小泉と杉原は、エッシャー風タイリング問題を扱いやすい最適化問題へと定式化を行い、その最適化問題の解を陽的に求めた。また、酒井は小泉の解法を改善する手法を提案した。それにより、質の良い解が得られることが確認されている。しかし、2 種類以上の図形に対するエッシャー風タイリング問題についてはあまり研究されていない。そこで、本研究では 2 種類の図形に対するエッシャー風タイリング問題についてアプローチを試みる。本稿ではまず、2 つの図形を入力して、それらを接合し、それに対し既存解法を適用する方法を提案する。さらに、入力図形を 1 つのみとし、もう一方の図形をあらかじめ用意しておいたデータベースから選び出す方法も提案する。

1 はじめに

タイリングとは、様々な図形によって、平面を隙間も重なりもなく敷き詰めたものである。タイリングは古くから建築物の装飾などに多く用いられており、現在においても広く使われている。タイリングによって作られる模様は単純な幾何学図形から、複雑な図形に至るまで多岐に及ぶ。タイリングには芸術的な側面の他に、多くの数学的な側面もあり、その繰り返しの中に現れる法則や性質、種類等が深く考察されている [3]。

オランダの版画家 M. C. Escher は数学的な見地からタイリングを研究し、タイリングに関する芸術的な作品を数多く残した [1]。Escher は動物などの形によるタイリングを作成しており、1 種類の図形を用いたものや、2 種類の図形を用いたもの、また、それ以上の複数種類の図形を用いた作品が存在する。

Escher の作品のようなタイリングを実際に作ろうとすると、芸術的なセンスや膨大な時間が必要となり、非常に難しいことがわかる。そこで、誰でもタイリングを作れるように、計算機を使ってタイリングを生成する問題が考えられる [4]。この問題は Escher の名前と作品にちなんで Escherization Problem (エッシャー風タイリング問題) [5] と呼ばれる。これは、入力図形が与えられたとき、その図形にできるだけ近く、かつ、平面を敷き詰めることができる図形を求める問題である。

1 種類の図形に対するエッシャー風タイリング問題については近年盛んに研究されてきた。小泉と杉原は、入力図形を多角形で与え、図形同士の近さを定める基準としてプロクラスティス距離

を導入することで最小化問題として定式化を行い、その解を陽的に求めることに成功した [9],[10]. また、酒井 [11] は、制約条件を緩和することによる、小泉らの解法の改善手法を提案した。それによって、1種類の図形によるタイリングの生成に関しては、質の良い解が得られることが確認された。

しかし、2種類以上の図形によるタイリングを扱った研究はあまり行われていない [6]. そこで本稿では、2種類の図形によるタイリングを扱い、そのときのエッシャー風タイリング問題に対してアプローチを試みる。

まず、本稿では、1種類の図形に対しての既存解法を利用した手法を提案する。すなわち、初めに入力図形として2つの図形を入力し、それらを接合して1つの図形とみなす。その図形に対して既存解法を用いてタイリングを生成し、最後に再び2つの図形に分離することで2種類の図形によるタイリングとする。次に、その実験結果より、入力図形の組み合わせが解の質に大きく影響を与えることを示す。そして新たに、図形の一方のみを入力として与え、もう一方をあらかじめ用意しておいたデータベースから自動的に選び出す問題について考え、その解法を提案する。

2 タイリングの基礎事項

この章では、タイリングの基礎的事項についてまとめる。より詳しくは文献 [4] を参照のこと。

2.1 諸定義

平面を有限種類の図形で隙間も重なりもなく埋め尽くしたパターンをタイリングといい、タイリングに使われる図形をタイルという。2つのタイルが交わるとき、交わりは曲線となり、その曲線をタイリング辺 (tiling edge) という。また、3つ以上のタイルが交わるとき、交わりは点となり、その点をタイリング頂点 (tiling vertex) という。1つのタイルに対し、タイリング頂点をつないでできる多角形をタイリング多角形 (tiling polygon) と呼ぶ。さらに、タイルが多角形のときは、タイリング辺やタイリング頂点と区別するために、多角形としての辺を図形の辺、多角形としての頂点を図形の点と呼ぶ。なお、以降では、単に辺と書くときはタイリング辺を指す。

これより、1種類のタイルからなるタイリングを考える。

タイリング上のタイルを1つ選ぶ。同じタイリング上のあるタイルに対し、元のタイルと重なるような合同変換のことを対称 (symmetry) という。タイリング上の任意の2つのタイルに対して対称が存在し、もしその変換を行ったときにタイリング全体も重なるのであれば、そのときその2つのタイルは同値 (equivalent) であるという。この同値関係によってタイリングのタイルは同値類に分類される。つまり、2つのタイルが同値であるとき、その2つのタイルは同じ同値類に分類されることとなる。タイリングが1つの同値類しか持っていないとき、すなわち、タイリング上のどの2つのタイルを選んでもそれらが同値であるとき、そのタイリングは isohedral タイリングと呼ばれる。

isohedral タイリングは、数学的に扱い易く、また、様々な形を表現するだけの柔軟性を持っている。実際に、M. C. Escher の1種類のタイルから成る作品のほとんどが、isohedral タイリングで出来ていて、さらにエッシャーの作成した2種類の図形によるタイリングでは、異なる2つのタイルを一つと見なすと isohedral タイリングとなることが知られている [5],[6]. したがって本研究では、タイリングの中でも、isohedral であるもののみを考えることとする。

2.2 isohedral タイリング

isohedral タイリングの性質は topological type と incidence symbol によって決定される。topological type とは isohedral タイリングの中の一つのタイルを取って、そのタイリング頂点において交わっているタイルの個数 (タイリング頂点の次数) を並べたリストのことを言い、isohedral の定義より、タイリングの中の全てのタイルでそのリストは同じである。topological type は、isohedral タイリングの場合、全部で 11 種類であることが知られている [3]。また、incidence symbol はタイルの隣接関係を制限するものであり、tile symbol と adjacency symbol を合わせたものである。incidence symbol は次のようにして得られる。

まず初めに、タイリングの中の任意のタイルを選んで、そのタイルの 1 つのタイリング辺に向き付きのラベル a を付ける。そして、今決めた向きにタイリング辺をたどり、ほかのタイリング辺にも同様に $b, c, \dots$ とラベルを付けていく (図 1 参照)。ただしこのとき、図 2 のように、タイリングが、タイルをそのタイル自身に移すような対称を持つ場合は、移る前と移った後のタイリング辺には、向きを含めて同じラベルを付けることとする。このようにして付けたラベルを並べたものが tile symbol である。図 1 の場合は $[a^+b^+c^+d^+e^+f^+]$ 、図 2 の場合は $[ab^+c^+dc^-b^-]$ となる。ラベルの右肩に付いている符号は、初めに決めた向きとラベルが同じ向きなら $+$ 、逆向きなら $-$ として決められた符号である。ただし、ラベルが両方の向きを持つ場合は、何も付けないこととする。

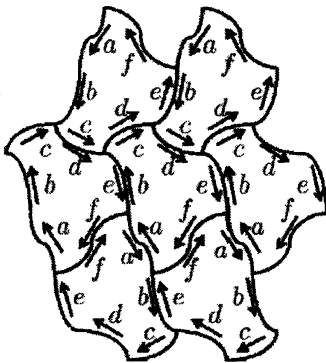


図 1: incidence symbol

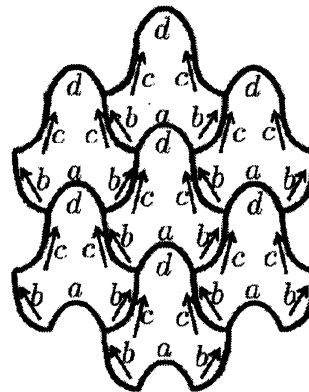


図 2: incidence symbol (鏡映を含む場合)

次に、得られたラベルを隣接する他のタイルにも付ける。そして、元のタイルのラベルを並べた順番に、元のタイルのラベルと隣接するラベルを並べたものが adjacency symbol である。ただし、対称性のある部分は省略することとする。図 1 の場合は $[a^+e^+d^-c^-b^+f^+]$ となり、図 2 は $[dc^-b^-a]$ となる。ここで、ラベルの右肩に付いている符号は、タイリング辺の両側にあるラベルが違う向きなら $+$ 、同じ向きなら $-$ として決められた符号である。

tile symbol と adjacency symbol を並べたものを incidence symbol とする。図 1 の場合は $[a^+b^+c^+d^+e^+f^+; a^+e^+d^-c^-b^+f^+]$ となり、図 2 の場合は $[ab^+c^+dc^-b^-; dc^-b^-a]$ となる。

topological type と incidence symbol によって、isohedral タイリングは、93 種類に分類されることが知られている [3]。これらは、それぞれ IH01, IH02, ..., IH93 と書いて区別される。

次に、タイルを変形することを考える。タイルの変形とは、タイリング頂点の位置を変えること

と、タイリング辺を変形するということであり、許される変形は incidence symbol によって制限される。まず、タイリング辺の変形について見る。タイリング辺の変形に対する制限は、incidence symbol から得られ、次の 4 種類に分けられる:

S 型 タイリング辺の両側のラベルが同じ名前で向きが異なるとき。この場合は、そのタイリング辺の midpoint に関して点対称でなければならない。その条件を満たす範囲で変形できる。

U 型 タイリング辺の両側の名前が異なり、少なくとも一方が向きを持たないとき。この場合は、そのタイリング辺の垂直二等分線に関して線対称でなければならない。その条件を満たす範囲で変形できる。

J 型 両側のラベルの名前が異なり、どちらも向きを持つとき。この場合は、そのタイリング辺は自由に変形できる。

I 型 両側のラベルの名前が等しく、どちらも向きを持たないか、どちらも同じ向きを持つとき。この場合は、S 型と U 型の両方の条件を満たさないといけなため、変形できない。すなわち、このタイリング辺は直線分でないといけな。

同じラベルが付いたタイリング辺は、全て同じ形になるということにも注意する。

タイリング頂点に関する制限も同様に、incidence symbol を見ることによって得られる。タイリング頂点を動かすということは、タイリング多角形の形を変えるということであるが、incidence symbol によって、タイリング多角形の辺の長さや、内角が制限されている。詳細については [5] において考察されている。

isohedral タイリングには 93 種類のタイプが存在するが、対称性の高いものは変形の自由度が少なく、エッシャー風タイリングを生成する際に、タイルを変形するという観点から見れば、その実行可能解がほかのタイプの実行可能解にすべて含まれてしまう場合がある。したがって、エッシャー風タイリングを生成する際には、対称性の高いものについては省いて考えることができる。いずれかの辺が変形可能であり、他のどのタイプにも含まれないものは 28 種類であることが知られているため [4]、本研究では 28 種類のタイリングタイプを考慮する。

3 既存解法

この章では 1 種類の図形に対するエッシャー風タイリング問題について述べる。そして、その問題に対する、小泉と杉原の定式化および解法 [9],[10] を紹介し、この手法の特徴についての考察を行う。また、その改良手法として酒井の手法 [11] を紹介する。

3.1 1 種類の図形に対するエッシャー風タイリング問題

1 種類の図形に対するエッシャー風タイリング問題とは、ある入力図形 S が与えられたとき、

1. 図形 T は平面を敷き詰めることができる、
2. 図形 T はできるだけ S に近い形である、

という2つの条件を満たす図形 T を見つける問題である [5]. この問題を最適化問題として書き換える. まず, 1つ目の条件は, 本稿では isohedral tiling のみを扱うので, 「図形 T は incidence symbol の定める制約に従う.」と書き換えることができる. 次に, 2つ目の条件は, 二つの図形を引数に取り, 距離の公理をみたすような関数 $d(S, T)$ を用いて, 「図形同士の距離 $d(S, T)$ が出来るだけ小さい.」と書き換えることができる. よって, この問題は以下のような最適化問題ということができる.

入力 図形 S ,
出力 図形 T ,
最小化 $d(S, T)$,
制約条件 incidence symbol の定める制約.

次節より, この問題に対する小泉らの定式化 [9],[10] を紹介する.

3.2 図形同士の距離

図形同士の距離 d は, 図形の表現方法や, 比較の仕方によって様々なものが考察されている. タイリングを扱う際は, 回転・伸縮・平行移動によって互いに重なる図形は同じ形であるとみなすため, 回転・伸縮・平行移動に関して不変な距離が望ましい. 小泉らはそのような特徴をもつ距離の一つである, プロクラステス距離 [8] を用いた.

プロクラステス距離を用いるために, まず図形を n 角形で与え, その n 個の頂点の x, y 座標を順番に並べた $2 \times n$ 行列 U によって図形を表現する. すなわち n 角形の頂点の x 座標を $P_{1x}, P_{2x}, \dots, P_{nx}$, y 座標を $P_{1y}, P_{2y}, \dots, P_{ny}$ としたとき, その図形は,

$$U = \begin{pmatrix} P_{1x} & P_{2x} & \cdots & P_{nx} \\ P_{1y} & P_{2y} & \cdots & P_{ny} \end{pmatrix}$$

という行列で表現される. ただし, 行列 U は, 第一頂点 P_1 を n 角形のどの点とするかによって変化することに注意する. また, 以下では, n 角形の頂点を $P_1, P_2, \dots, P_n$ としたとき, その位置ベクトルも同じ記号 P_i で表すこととし, 図形とその図形から得られる行列も, 同じ記号で表すこととする.

重心が原点に重なるように配置された図形 U, W に対して, プロクラステス距離 $d(U, W)$ は次式で定義される.

$$\begin{aligned} d^2(U, W) &= \min_{s, \theta} \left\| sR(\theta)U - \frac{W}{\|W\|} \right\|^2 \\ &= 1 - \frac{\|UW^\top\|^2 + 2\det(UW^\top)}{\|U\|^2\|W\|^2}. \end{aligned} \tag{1}$$

ここで, $\|X\|$ は行列 X のフロベニウスノルム, s はスカラー, $R(\theta)$ は θ ラジアン of 回転行列を表す. この距離は定義から回転・伸縮・平行移動に関して不変である.

3.3 定式化

3.1 節の最適化問題を定式化する. まず, 目的関数について考える. 重心が原点と重なるよう配置された入力図形を W , 求める図形 (タイル) を U とする. (1) 式より, W が与えられたときに

この2つの図形の距離 $d(U, W)$ を最小にすることは、次式を最大にすることとなる：

$$\frac{\|UW^\top\|^2 + 2\det(UW^\top)}{\|U\|^2}. \quad (2)$$

ここで、 $\mathbf{u}_x$ を図形 U の各頂点の x 座標を並べた n 次元縦ベクトルとし、同様に、 $\mathbf{u}_y$ を図形 U の y 座標を並べたベクトル、 $\mathbf{w}_x$ を図形 W の x 座標を並べたベクトル、 $\mathbf{w}_y$ を図形 W の y 座標を並べたベクトルとして、

$$U = \begin{pmatrix} \mathbf{u}_x^\top \\ \mathbf{u}_y^\top \end{pmatrix}, W = \begin{pmatrix} \mathbf{w}_x^\top \\ \mathbf{w}_y^\top \end{pmatrix}, \mathbf{u} = \begin{pmatrix} \mathbf{u}_x \\ \mathbf{u}_y \end{pmatrix}, \mathbf{w} = \begin{pmatrix} \mathbf{w}_x \\ \mathbf{w}_y \end{pmatrix}$$

と表す。また、対称行列 V を

$$V = \mathbf{w}\mathbf{w}^\top + E\mathbf{w}\mathbf{w}^\top E^\top \quad (3)$$

とする。ただし、 $E = \begin{pmatrix} O & I \\ -I & O \end{pmatrix}$ であり、 O は零行列、 I は単位行列である。これらを用いると、(2) 式は、次の形式で書くことができる：

$$\frac{\mathbf{u}^\top V \mathbf{u}}{\mathbf{u}^\top \mathbf{u}}. \quad (4)$$

次に制約条件について考える。これはタイリングタイプによって異なるが、ここでは例として IH07 について述べる。IH07 の incidence symbol は、図 3 で与えられる。ここで、図 4 のように点を取ることにする。このとき点の数を n とし、 $N = n/6$ とする。incidence symbol から分かる

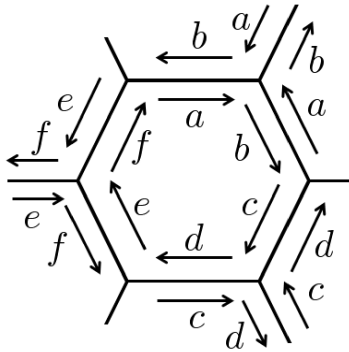


図 3: IH07 の incidence symbol

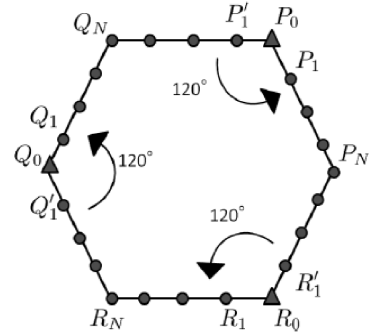


図 4: 境界上の点の関係

ように、すべての辺は J 型であり、同じ辺上の点同士の間には制約がない。また図 3、図 4 から、 $\angle P_0, \angle Q_0, \angle R_0$ を挟む 2 辺は同じラベルを持つ。よって、タイリング辺は P_0, Q_0, R_0 を中心とする 120 度回転で重ならない。したがって S を 120 度回転を施す行列とすると次式が成り立つ：

$$\begin{cases} S(P'_i - P_0) = P_i - P_0 & (i = 1, \dots, N) \\ S(Q'_i - Q_0) = Q_i - Q_0 & (i = 1, \dots, N) \\ S(R'_i - R_0) = R_i - R_0 & (i = 1, \dots, N). \end{cases} \quad (5)$$

IH07の場合は回転のみを考慮したが、同様に考えることにより、ほかのタイプで出てくる平行移動・並進鏡映等で重なる条件も定式化できる。

(5) 式には定数項が入っておらず、すべて変数の線形結合だけで表されている。したがって、行列 A を用いて、

$$Au = 0 \quad (6)$$

とまとめることができる。同様にすべての isohedral タイリングのタイプについて制約条件を (6) 式の形で書くことができる（行列 A のサイズはタイプによって異なる）。

3.4 固有値問題への帰着

3.3 節より、目的関数は (4) 式、制約条件は (6) 式に定式化された。したがってエッシャー風タイリング問題は

$$\begin{array}{ll} \text{最大化} & \frac{\mathbf{u}^\top V \mathbf{u}}{\mathbf{u}^\top \mathbf{u}}, \\ \text{制約条件} & A\mathbf{u} = 0 \end{array} \quad (7)$$

という最適化問題と書くことができる。ここで注意が必要なのは、行列 A が isohedral タイリングのタイプによって異なることと、入力図形 W の第一頂点の決め方によって、行列 V が変わることである。つまり、(7) は isohedral タイリングのタイプを 1 つに決め、入力図形 W の第一頂点を固定したうえでの問題である。実際は、全ての isohedral タイリングのタイプについて、第一頂点の点を変えながら最適化問題 (7) を解き、全ての場合を計算したのちに、最も目的関数値が大きいものを最適解とすることとなる。

ここで、制約条件を変形する。Ker A の正規直交基底を $\mathbf{b}_i$ ($i = 1, \dots, m$) とすると、制約式 (6) は、

$$\mathbf{u} = \xi_1 \mathbf{b}_1 + \xi_2 \mathbf{b}_2 + \dots + \xi_m \mathbf{b}_m \quad (8)$$

と書ける。行列 B を Ker A の正規直交基底を並べたもの、 $\boldsymbol{\xi}$ を $\xi_1, \xi_2, \dots, \xi_m$ を並べた縦ベクトルとすると、(8) 式は、

$$\mathbf{u} = B\boldsymbol{\xi} \quad (9)$$

と書ける。以上を用いると、最適化問題 (7) は、

$$\text{最大化} \quad \frac{\boldsymbol{\xi}^\top B^\top V B \boldsymbol{\xi}}{\boldsymbol{\xi}^\top \boldsymbol{\xi}} \quad (10)$$

という、無制約最適化問題となる。これは $\boldsymbol{\xi}$ のレイリー商最大化問題であり、対称行列 $B^\top V B$ の最大固有値を求める問題である。この行列の最大固有値に対応する固有ベクトル $\boldsymbol{\xi}$ から、出力図形を表すベクトル $\mathbf{u} = B\boldsymbol{\xi}$ が求まる。

固有値問題の解法としては、射影法 [12] を用いる。(3) 式より、行列 V はランク 2 の対称行列であるため、行列 $B^\top V B$ はランクが高々 2 の対称行列である。したがって、 $B^\top V B$ の非零の固有値に対応する独立な固有ベクトルは高々 2 本であり、それらによって張られる空間を考えることで、 2×2 行列の固有値問題に落とすことができる。これにより、固有値問題の解を陽的に求めることができる。

3.5 制約条件の緩和

小泉らの解法は、陽的に解を求めることができるものの、形の似た入力図形でも点の配置によって解の質が変わるという問題点がある。その原因は制約条件にある。小泉らの提案した制約条件は、図4や(5)式を見ると分かるように、各タイリング辺上の点の数が等しくなければならない。つまり、入力の多角形において、タイリング頂点に対応する図形の点が1つ決まると、残りのタイリング頂点に対応する点も自動的に決まることになり、タイリング頂点の位置に関する自由度が低いと言える。そこで、酒井は制約条件を定式化する際に、タイリング辺上の点の数に自由度を持たせることで制約条件を緩和する方法を提案した[11]。

3.3節と同様に IH07 を例にとって考える。図4のタイリング辺上の点の数に自由度を持たせると、図5のようになる。

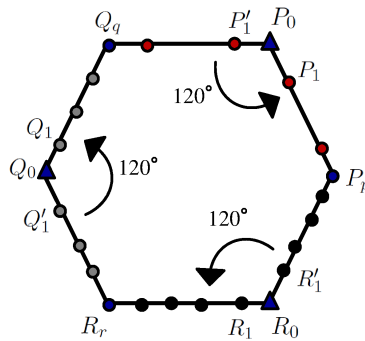


図 5: タイリング辺上の点の数に自由度を持たせた図形

このとき、(5) 式は次のように変化する。

$$\begin{cases} S(P'_i - P_0) = P_i - P_0 & (i = 1, \dots, p) \\ S(Q'_i - Q_0) = Q_i - Q_0 & (i = 1, \dots, q) \\ S(R'_i - R_0) = R_i - R_0 & (i = 1, \dots, r). \end{cases} \quad (11)$$

この式は(5)式と同様に、 $\{A_k u = 0 \mid 1 \leq k \leq (p, q, r) \text{ の組合せ} \}$ の形で書くことができる。 p, q が決まると r は一意に決まることに注意すれば、 k の数は $O(n^2)$ となる。この中で目的関数が最大となるタイルを求めればよいが、タイリングタイプによっては k の数が $O(n^3)$ や $O(n^4)$ となるものもある。行列 A から行列 B を求める計算の計算量は $O(n^3)$ であり、すべてのタイルを求めると計算時間が膨大となってしまう。そこで、ある程度質の良いタイルを近似的に求めることとし、その手法としては局所探索法を用いる。

局所探索法とは、ある解に少しの変更を加えて、元の解よりも良くなったとき、変更後の解に移動するという操作を、それ以上良い解が見つからなくなるまで繰り返す方法である。つまり、適当な解 x から始め、 x の近傍 $N(x)$ に改善解 x' が存在する場合、 $x := x'$ とするという操作を、近傍内に改善解がなくなるまで繰り返す。

局所探索法を用いる際には近傍を定義する必要がある。ここでは近傍 $N(A)$ を、タイリング辺のいずれかの辺上の点を1つ増やしたときの制約条件行列 A の組合せとする。つまり、IH07 の場

合は、タイリング辺上の点の数を (p, q, r) とすると、近傍の全組合せは以下のようになる。

$$\begin{aligned} & (p+1, q-1, r), (p+1, q, r-1), (p-1, q+1, r), \\ & (p, q+1, r-1), (p-1, q, r+1), (p, q-1, r+1). \end{aligned}$$

このようにして得られた近傍を用いて局所探索を行う。タイリング辺上の点の数をすべて等しくしたときの制約条件行列を $A^{(1)}$ とし、 $A^{(1)}$ から得られた解を初期解とする。次に、上で述べた近傍内の制約条件行列 $A \in N(A^{(1)})$ をランダムに得て目的関数値を求める。ここで目的関数値が大きくなるようなタイルが得られたときそれを暫定解とする。この操作を繰り返していき、近傍内の全ての制約条件行列 A に対して暫定解の更新が行われなくなったとき、その解を出力する。

この手法を用いて数値実験を行った結果を図 6 に示す。(a) は入力図形、(b) は小泉らの手法で求まるタイル、(c) は酒井の手法で求めたタイル、(d) はそのタイリングである。

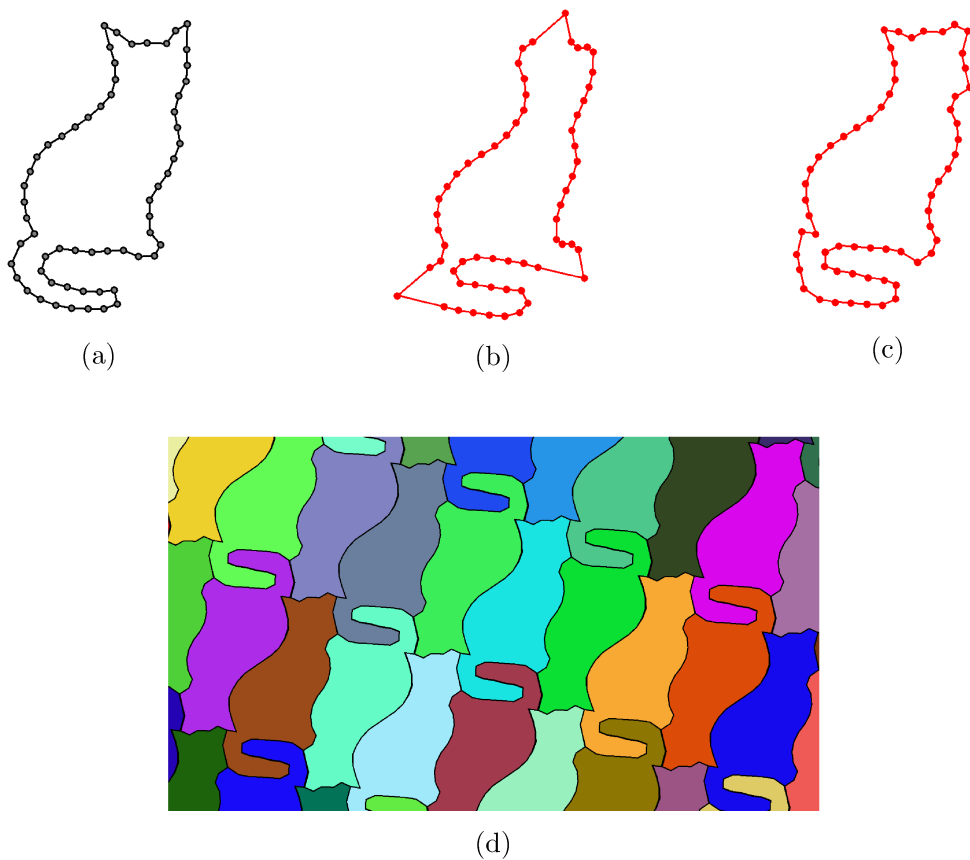


図 6: ネコのタイリング

図 6 より、制約条件の緩和による解精度の向上が確認できる。ただし、酒井の手法は局所探索法に基づいているため、小泉らの手法よりも計算時間がかかる。図 6 の実験では、(b) に約 0.6 秒かかったのに対し、(c) には約 13.0 秒かかり、およそ 20 倍の時間がかかることが分かる。

4 2種類の図形によるタイリング生成

この章では、本研究で扱う2種類の図形に対するエッシャー風タイリング問題について説明し、図形の接合を用いた手法を提案する。

4.1 2種類の図形に対するエッシャー風タイリング問題

2種類の図形に対するエッシャー風タイリング問題は、2つの入力図形 S_1, S_2 が与えられたとき、

1. 図形 T_1, T_2 はその2つで平面を敷き詰めることができる。
2. 図形 T_1, T_2 はできるだけ S_1, S_2 に近い形である。

という2つの条件を満たす図形 T_1, T_2 を見つける問題となる [6]。この問題も 3.1 節と同様、最適化問題に書き換えると、

入力	図形 S_1, S_2 ,
出力	図形 T_1, T_2 ,
最小化	$d(S_1, T_1), d(S_2, T_2)$,
制約条件	T_1 と T_2 で平面の敷き詰め可能,

となる。

4.2 提案手法

本研究では1種類の図形に対する既存解法を用いることを考え、次の流れでタイリングを生成する。まず、2つの入力図形を接合して1つの図形とする。次に既存解法を用いて、1つの図形としてタイリングを生成する。最後に、タイルを再び2つに分けることで、2種類の図形によるタイリングとする。



図 7: 2種類の図形によるタイリング生成のイメージ

このような方法を使う上で、簡単のため、前節の目的関数を変更する。図7のように、接合の際に変形した後の S_1, S_2 をそれぞれ S'_1, S'_2 、接合した図形を S 、それに既存解法を用いて得られた図形を T とし、

$$\text{最小化} \quad d(S_1, S'_1), d(S_2, S'_2), d(S, T),$$

とする。以降では、図形の接合方法と全体のアルゴリズムについて説明する。

まず、図形の接合方法について述べる。図形は既存解法と同様に、 n 角形で与えることとする。初めに、2つの入力図形についてそれぞれ接合部分を設定し、その端点が重なるように拡大・縮小、回転、平行移動を施す。ただし、この段階では、2つの図形の間に隙間や重なりが生じる。そこで

次に、それを避けるために図形を変形する。しかし、大きな変形はタイリングの質の低下につながる。そこで、図8のように、変形が最も小さくなる位置に図形をずらす。最後に2つの図形の間を通る線(分離線)を引き、図形を接合したこととする。分離線は後で再び図形を分離するとき用いる。詳しい接合方法は文献[7]を参照のこと。

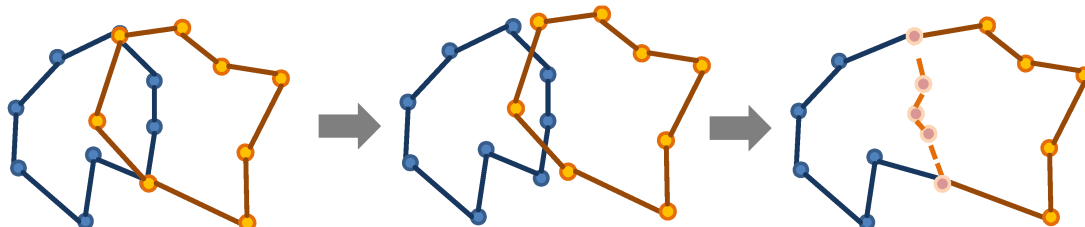


図 8: 2つの入力図形の接合方法

次に、全体のアルゴリズムについて述べる。まず、2つの図形を入力し、それぞれの図形の接合部分を設定する。次に、図形を接合する。この時点で拡大・縮小などに制限を設け、制限を満たさない場合や図形に交差が生じた場合は計算を打ち切り、新たな接合部分を設定し、再び計算を行う。次に、接合された図形に対して既存解法を用いてタイリングを生成する。その時点で暫定解が存在する場合は、両者を比較し、より良い解が得られた場合は暫定解を更新する。タイリングの評価には、接合の際にどれだけ変形されたかも考慮に入れる。そして、図形の接合に戻り、新たな接合部分を設定して再び計算を行う。この操作を全ての接合部分の組合せについて行った後に、最適解を出力する。

既存解法として、小泉らの手法と酒井の手法を紹介した。酒井の手法を用いれば、より良い解が得られると期待できるが、酒井の手法は小泉らの手法と比べ、約20倍の計算時間がかかるという問題がある。そこで、上のアルゴリズム中のタイリング生成には小泉らの手法を用いることとする。そして、一番良い解のみでなく、良い解が得られたときの接合後の図形を複数保存することとし、全ての接合後の図形に対して小泉らの手法を用いて評価した後で、保存されている図形に対してのみ酒井の手法を適用し、その中の最良解を出力する。

4.3 数値実験と考察

4.2節で述べた手法を用いて数値実験を行った結果を示す。図9と図10において、(a)は入力図形、(b)は出力図形、(c)はそのタイリングを表す。計算機環境は表1の通りである。計算時間は、10分から15分程度であった。

表 1: 計算機環境

CPU	Intel Core i7-4770 (3.40GHz)
メモリ	8GB
OS	Windows 7
言語	Java

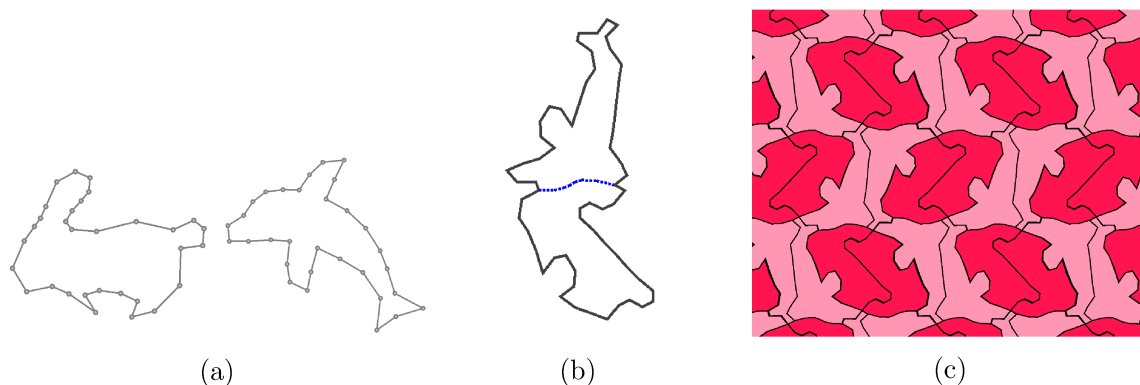


図 9: ウサギとイルカのタイリング

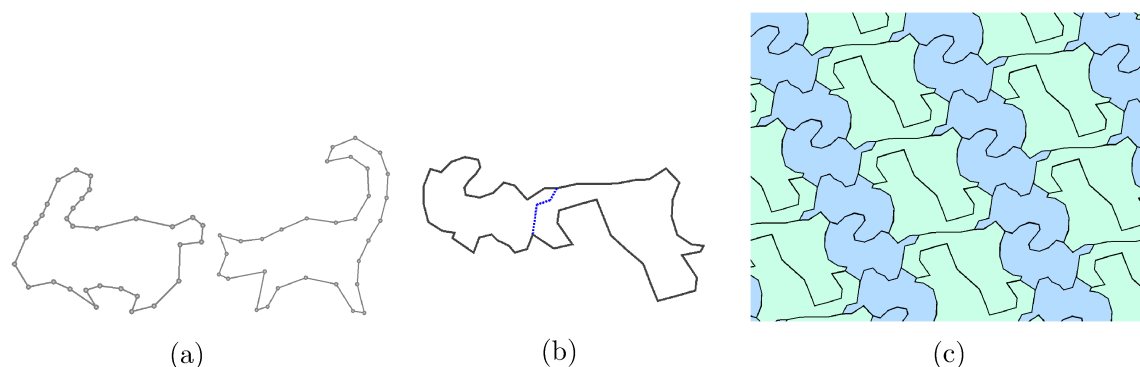


図 10: ウサギとネコのタイリング

以上の結果について考察を行う。図9と図10を比べると、図9の方がよりきれいなタイリングが得られたと感じられる。実際、図9の評価値は0.066、図10の評価値は0.078で、評価値からも図9の方が良い結果が得られていると分かる。これより、一方の入力図形は等しいにもかかわらず、もう一方の図形が異なることによって、得られる解の質が変わることが分かる。このことから、2つの入力図形の組み合わせが重要であると考えられる。

入力図形の形が解の質に影響を与えるというのは、1種類の図形に対する問題においても言えることである。しかし、1種類のみを図形を扱う場合はこの問題点に対応する方法が考えにくいのに対し、2種類の図形を扱う場合は、相性の良い組み合わせを選ぶことができれば解決できると考えられる。そこで、次章より、相性の良い入力図形の組み合わせを見つけることについて考える。

5 一方の図形を自動選択する問題

4章で提案した手法において、2つの入力図形の組み合わせが解の質に大きく影響を与えることより、この章では、入力図形を1つのみとし、もう一方の図形として相性の良さそうな図形を自動的に選び出す手法を考える。そのためにまず、扱う問題を新たに定義し直し、図形の選択方法やアルゴリズム等について提案を行う。

5.1 問題の再定義

2つの図形のうち一方のみを入力とするため、以下の問題を考える。まずあらかじめ、データベースとして図形を多数（数十から数百）用意しておく。次に、入力として図形を1つ与える。そして、その図形と相性の良さそうな図形をデータベースから選び出す。最後に、入力図形と選ばれた図形の2つを用いてタイリングを生成する。これを最適化問題として書くと、

入力 図形 S_1 , 図形のデータベース,
出力 図形 S_2, T_1, T_2 ,
最小化 $d(S_1, T_1), d(S_2, T_2)$,
制約条件 T_1 と T_2 で平面の敷き詰め可能,

となる。次節より、この問題に対する手法を提案する。

5.2 提案手法

図形の選択方法、タイリングの生成方法、及び全体のアルゴリズムについて提案を行う。

5.2.1 図形の選択方法

まず、入力図形が1つ与えられたとき、その図形と相性の良さそうな図形をデータベースから選ぶ方法について述べる。

入力図形を規則的に並べると、図 11 のように同じ形の隙間が多数できると考えられる。本研究では、その隙間の形とデータベースの図形の形を一つずつ比較していき、最も近かったものを2つ目の図形とするという発想を用いる。そのためには、どのようにして図形を並べるか考える必要がある。

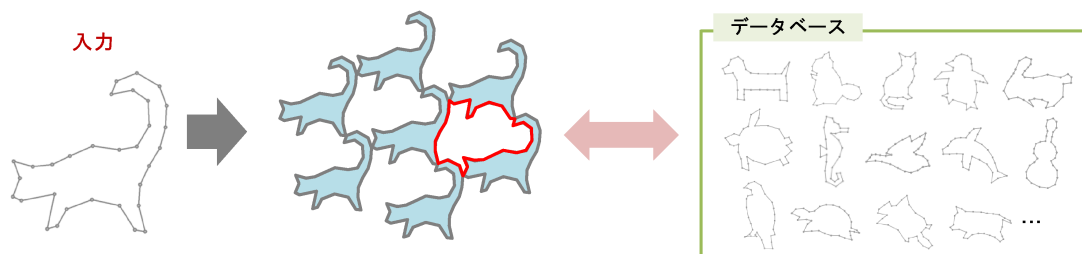


図 11: 図形選択のイメージ

最終的なタイリング生成に関しては、4章と同じく既存解法 [9],[10],[11] を用いるため、平面の敷き詰め可能な条件は、2つの図形を接合したものに対する isohedral タイリングの定める制約となる。そこで、図形の並べ方は isohedral タイリングをもとにする。つまり、isohedral タイリングの定める制約を満たすタイルをどこかで2つに分割し、その片方に入力図形を入れるといった考え方である。タイルの分割は、簡単のためタイリング頂点を結ぶ線で行う。それでも、例えば1つのタイルに対しタイリング頂点が6つのタイプの場合、分割する線の引き方は9通りあり、更にそれぞれに対し、どちらの領域に入力図形を入れるかの2通りがある。それぞれのパターンに対

して計算を行う必要があるが、タイプによっては、そのうちのいくつかが等しくなり、実際に計算するパターン数は少なくなることが多い。

例として、IH01 の場合について説明する。incidence symbol は、図 12 で与えられる。すべての辺が J 型で、各対辺が同じラベルを持つ。つまり IH01 は、各対辺が平行移動で重なるという制約を持つタイプである。この制約は非常にシンプルであるため、実際に計算するパターンは 3 つしかない。それぞれについて、具体的な図形の並べ方を説明する。

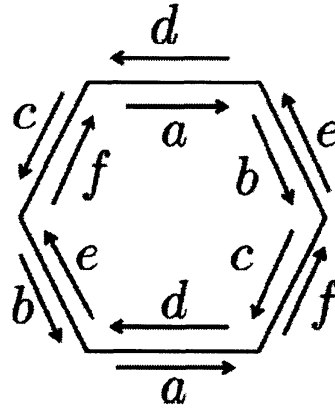


図 12: IH01 の incidence symbol

1 つ目のパターンは、図 13 のように、六角形を 2 つの四角形に分割したものである。この図において、同じ印の付いた辺は同じ形であることを意味する。IH01 の場合はどの線で分割してどちらの領域に入力図形を当てはめても同じ条件となるため、1 つのみ考えればよい。

IH01 のタイルを敷き詰めると図 14 のようになる。色を塗った部分がパターン 1 で入力図形を入れる部分である。その部分について見ると、タイリング頂点の 4 点が重なるように並んでいることが分かる。このことから、入力図形において、タイリング頂点に対応する点を 4 つ選び、それらが重なるように並べていくことで、隙間の形を得ることができる。図 15 にその例を示す。星印はタイリング頂点の位置を表す。(以降の図についても同様に、星印はタイリング頂点を表す。) 図 15 の星印を直線で結べば、図 14 と対応していることが分かる。

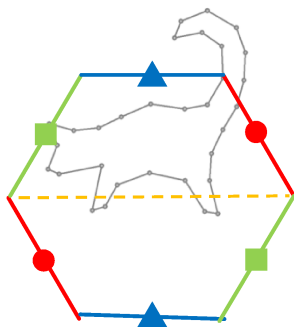


図 13: パターン 1 のタイルの分割

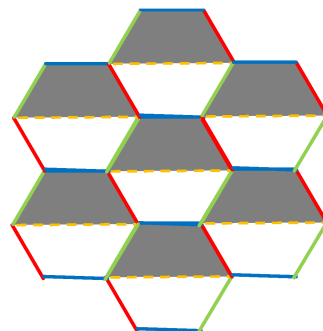


図 14: パターン 1 のタイリング

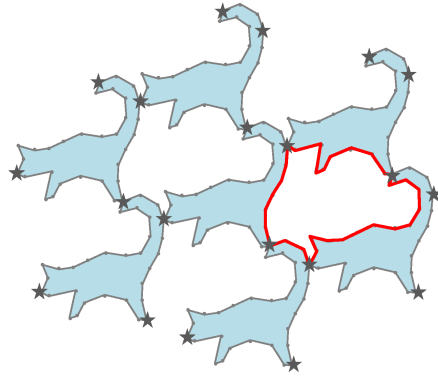


図 15: パターン1の並べ方

図 15 では入力図形をきれいに並べることができたが、実は、タイリング頂点の選び方によっては、入力図形を並べた際に入力図形同士が重なることがある。その例を図 16 に示す。タイリングでは重なりは認められないため、そういった場合は計算を打ち切り、新たなタイリング頂点を設定することとする。

実際の計算においては、隙間の形さえ得られれば良いので、入力図形を実際に並べることはせず、並べたと仮定したときの隙間の形を求める。入力図形が重なる場合は、隙間の形を作ったときに、その形に自己交差が生じることとなる。図 16 の場合は図 17 が隙間の形として求まり、交差が生じていることが分かる。そのため、入力図形の重なりを判定するには、得られた隙間の形の交差判定を行えばよい。本研究ではその方法として、隙間の形も多角形として得られるため、その多角形の各辺が、隣接する辺を除く他のすべての辺と交差を持つかを調べる方法を用いた。以降のパターンにおいても、得られた隙間に対して交差判定を行う。

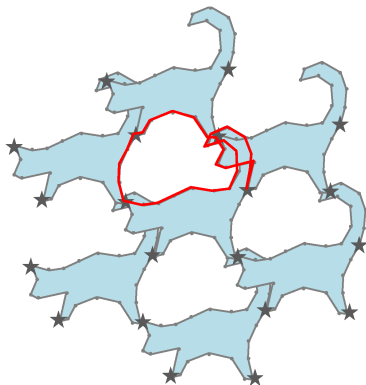


図 16: 入力図形が重なる場合

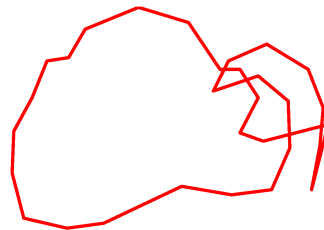


図 17: 入力図形が重なったときの隙間の形

2 つ目のパターンは、図 18 のように、六角形を三角形と五角形に分割し、五角形側に入力図形を当てはめたものである。この分け方についても、IH01 の場合はどの線で分割しても同じ条件となるため、1 つのみ考えればよい。このパターンでは、入力図形からタイリング頂点を 5 つ選ぶが、このとき、丸印の辺が五角形側に 2 つあるので、そこに対応する 2 辺上の点の数は等しくなるようにする。

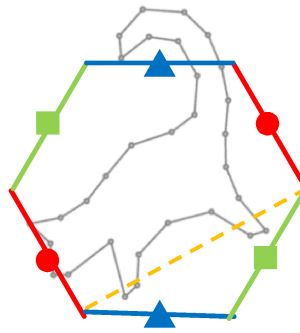


図 18: パターン 2 のタイルの分割

入力図形を入れる領域に丸印の辺が2つあるということは、その2辺に対応する部分は同じ形で平行移動で重ならなくてはならない。しかし、ほとんどの場合はそのようにはならないため、このままでは図形をきれいに並べることはできない。そこで、入力図形を変形する。その流れを図 19 に示す。丸印の辺に対応する2辺については、点の数が同じになるようタイリング頂点を取ってあるので、平行移動で重なるべき点の対応関係が分かっている。そこで、まず、各対応する点の中点を取っていくことで、辺を1つ作る。それを変形してしない部分に接合する。このとき、変形していない部分は2つに分けられているが、その2つをつなぐ部分を変形したことによって、角度や大きさのつじつまが合わなくなっており、そのまま接合することはできない。そこで、変形していない部分の一方を、形は変えずに拡大縮小、回転を施すことで、つじつまを合わせ、そのうえで接合する。以上のようにして図形を変形することにより、図 20 のように、図形を並べ、隙間の形を得ることができる。

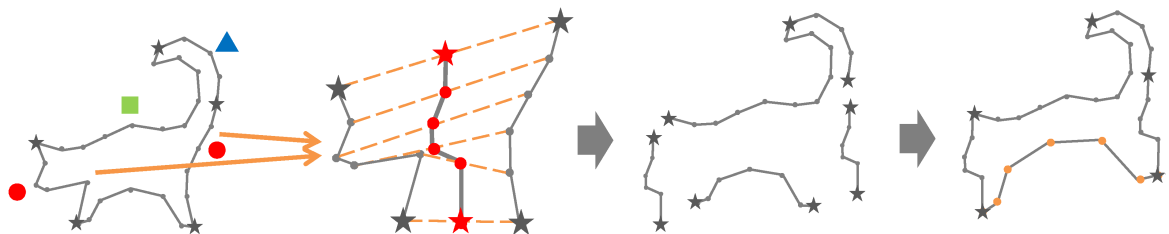


図 19: パターン 2 の入力図形の変形

最後に、3つ目のパターンについて述べる。図 21 のように、六角形を三角形と五角形に分割し、三角形側に入力図形を当てはめたもので、この分け方についても IH01 の場合はどこで分割するかによらないので、パターンは1つしかない。このパターンではタイリング頂点に対応する点を3つ選ぶ。これを並べるのだが、入力図形を当てはめただけでは丸印の辺の形が決まらないので、図 22 のように、入力図形が繋がらず、隙間の形に自由度が残った状態となる。

そこで、隙間の形をデータベースの図形を使って決める。まず、データベースから図形を1つ呼び出してくる。このとき、入力図形とデータベースの図形の点の数が同じ場合は、形が決まっている部分だけですでに図形の点の数に達してしまっているため、丸印の辺上の点の数を決め、その分だけデータベースの図形の点の数を増やす。後ほど本稿で示す実験では、14個増やしている。

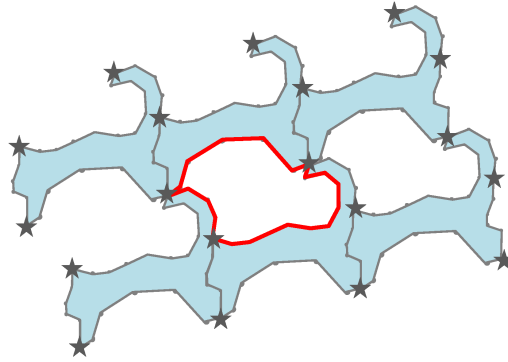


図 20: パターン 2 の並べ方

そして、この図形に対しても、タイリング頂点に対応する点を選ぶ。ただし、この時点で、各辺上の点の数は決まっているので、1 か所決めれば、他は自動的に決まる。そして、パターン 2 と同じように、丸印の辺に対応する 2 辺について、対応する点の中点を取っていくことで辺を 1 つ作り、それをすでに形が決まっている部分に、適切な拡大縮小・回転を施したうえで接合する。以上より、図 22 のように隙間の図形を決定することができる。このパターンでは、データベースから呼び出される図形が変わるごとに、また、呼び出された図形におけるタイリング頂点に対応する点が変わるごとに、隙間の形が変わることになる。

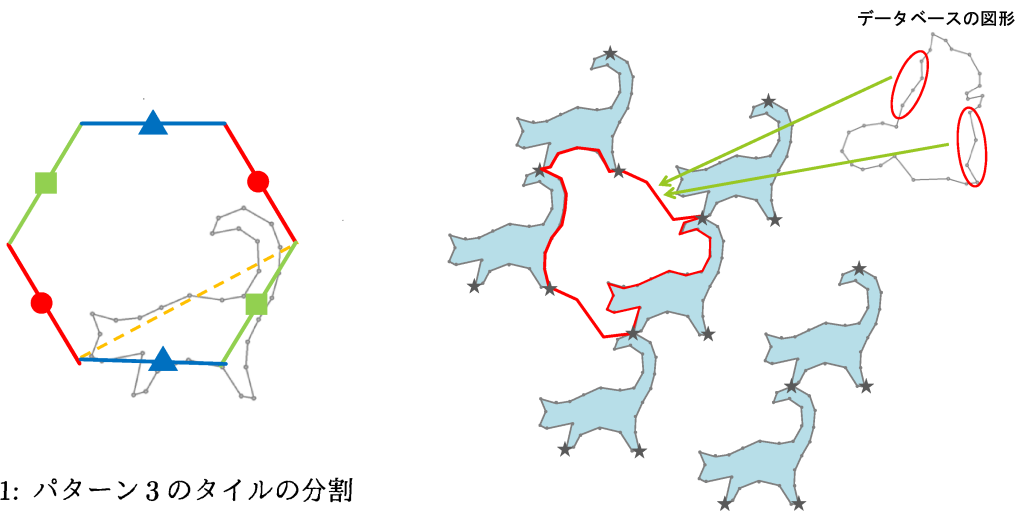


図 21: パターン 3 のタイルの分割

図 22: パターン 3 の並べ方

以上が、各パターンにおける、図形の並べ方と隙間図形の作り方である。IH01 について述べたが、他のタイプでも同じように考える。例えば、制約条件に回転が含まれるタイプであれば、パターン 2 で変形したときのような考え方で、回転によって重なるように変形すればよい。

隙間の形が決まったら、次にデータベースの図形を呼び出し（パターン 3 では既に呼び出されている）、1 つ 1 つ隙間の形とのプロクラスティス距離を計算する。この値が小さいものを採用する。ただし、パターン 2 では入力図形を変形したため、変形前と変形後の図形のプロクラスティス距離

も計算し、その2つを足したものを評価値とする。つまり、

$$\text{評価値} = (\text{入力図形と変形後の図形の距離}) + (\text{隙間図形とデータベースの図形の距離})$$

とする。

以上を、考える全てのタイリング頂点の選び方、全てのデータベースの図形に対して行い、評価値が小さいものを採用する。そこで選ばれた図形と入力図形を用いて2種類の図形によるタイリングを生成する方法を次節で述べる。

5.2.2 タイリング生成

2つ目の図形がデータベースから選ばれたのち、タイリングを生成する方法を述べる。図 15, 図 20, 図 22 において、白い部分を2つ目の図形と見ると、この段階でタイリングにはなっている。しかしこれでは、入力図形はほとんど変形されていないのに対して、もう一方の図形はかなり変形していることになり、バランスが悪い。そこで、新たにタイリングを生成し直す。

その方法としては、4章のような方法を使う。つまり、図形を接合して1つの図形と見て、それに対し、1種類の図形に対しての既存解法を用いる。今回の場合は、5.2.1 節で述べた図形を選択するための操作において、どの部分で接合すればよいという情報は得られているので、4章で述べたように接合部分を変えながら計算する必要はない。

既存解法としては、3.5 節で述べた酒井の手法を用いる。3.5 節では、局所探索法を用いて各辺上の点の数の探索を行ったが、今回の場合は、各辺上の点の数についても図形を選択するときに得られているので、局所探索を行う必要はない。

まとめると、データベースから図形を選択する際に得られた、接合部分の情報を用いて図形を接合し、各辺上の点の数の情報を用いて酒井の手法を一度のみ適用しタイリングを生成する。最後に図形を分離し、2種類の図形によるタイリングが完成する。

5.2.3 アルゴリズム

最後に、提案手法のまとめとして、全体のアルゴリズムについて述べる。その流れを図 23 に示す。

まず準備として、データベースの図形を多数用意しておく。これは、一度作っておけば、その後の計算において毎回使うことができる。

図形が準備できたら、計算に入る。初めに、入力として多角形を1つ与える。次に、タイリングタイプとそのパターン、タイリング頂点を設定する。その上で、5.2.1 節で述べた図形の並べ方を用いて、隙間の形を作る。ここで、隙間の形の交差判定を行い、交差が検出されたら計算を打ち切り、新たなタイリング頂点を設定する。また、入力図形の変形の際などに拡大縮小、回転に制限を設け、それを満たさない場合も同様に計算を打ち切る。

次に、隙間の形をデータベースの図形と比較する。評価値が良かった場合、その図形とタイリング頂点などの情報を保存しておき、後ほどタイリング生成に使うことになるが、保存するのは1つのみではなく、複数とする。これは、この段階での評価値と既存解法を用いたときの評価値が異なるため、1つのみとしてしまうと、既存解法の評価値の意味で良いものを逃してしまう可能性があるからである。つまり、保存数を m 個としたとき、評価値がこれまでの m 位のものよりも良かった場合に、その図形等を更新することになる。

この操作を，全てのタイリングタイプ，パターン，タイリング頂点の選び方に対して行う．さらに，入力図形を反転させたものについても行う（IH21 というタイプに関しては，制約条件が左右対称でないため，さらにデータベース側の図形を反転させたものに対しても行う）．その後，最終的に保存されている m 個の候補に対して，5.2.2 節で述べたように，酒井の手法によりタイリング生成を行い，評価値が最も良いものを出力する．

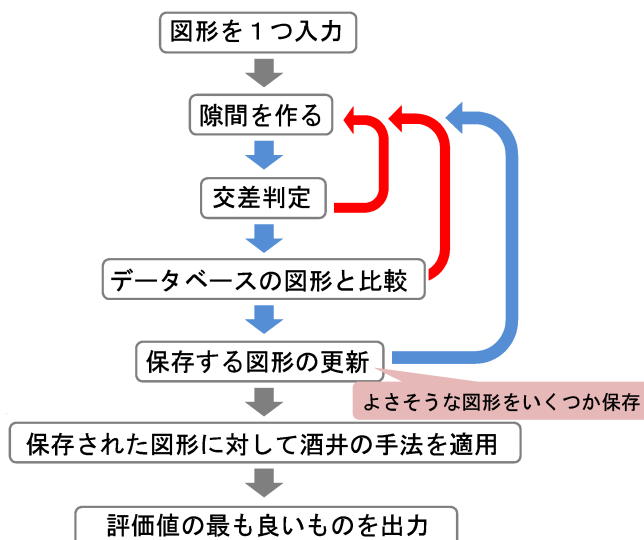


図 23: 一方の図形を自動選択しタイリングを生成する流れ

5.3 数値実験

5.2 節で述べた提案手法を用いて数値実験を行った結果を示す．計算機環境は 4.3 節と同様である．データベースの図形の数 は 50 個，図形の選択において保存する個数は 10 個とした．ただし，この 10 個の中には，同じ図形でタイリング頂点の位置などが異なるものも含まれる．また，isohedral タイリングのタイプは，IH01, IH04, IH21 のみで行った．

図 24 から図 26 において，(a) は入力図形，(b) は選ばれた図形と酒井の手法適用前の入力図形の変形とその隙間，(c) は酒井の手法適用後に得られた出力，(d) はそのタイリングである．

また，表 2 に各実験における計算時間を示す．この表において，図形選択は図形入力後からもう一方の図形の候補を選択するまでにかかった時間，すなわち図 23 において，ループが終了するまでの時間を表す．また，タイリング生成は，ループ終了後から解が出力されるまでの時間を表す．

表 2: 計算時間 (秒)

	図形選択	タイリング生成	合計
ネコ	108.71	2.54	111.25
魚	114.54	1.50	116.04
ゾウ	166.58	2.52	169.10

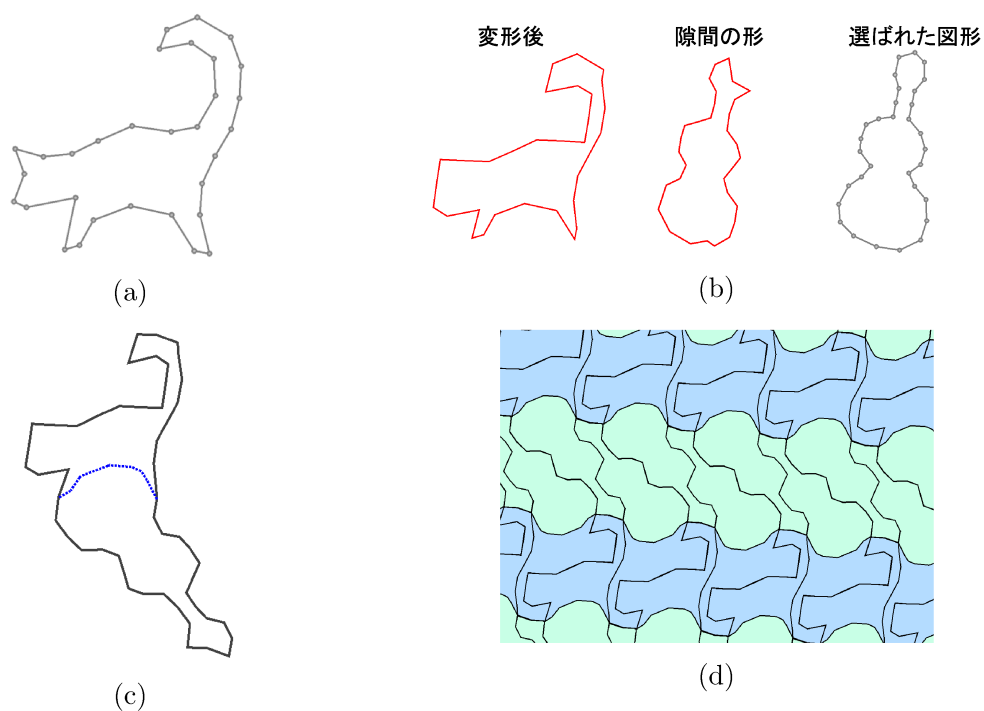


図 24: ネコの実験結果

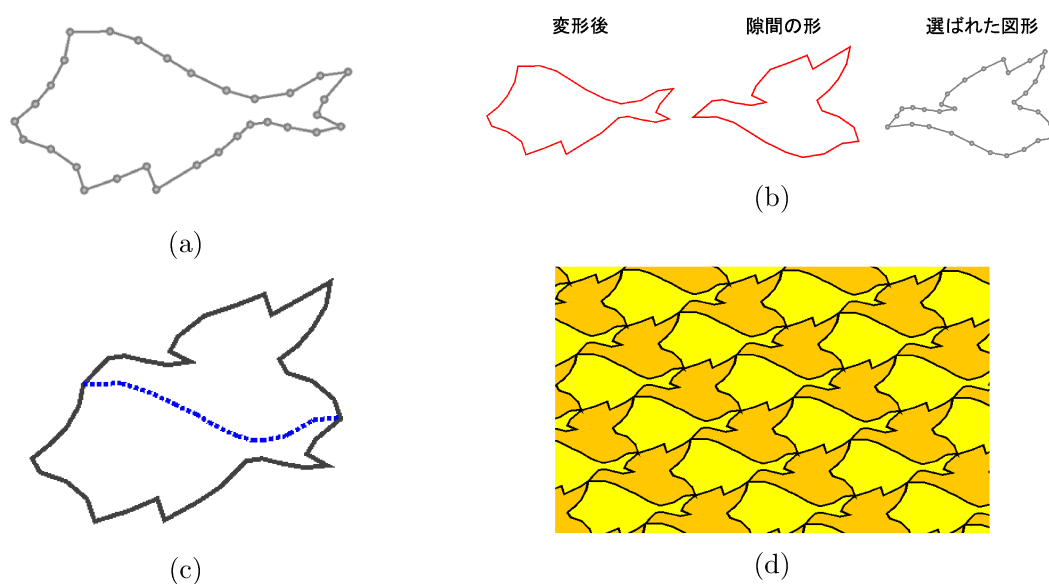


図 25: 魚の実験結果

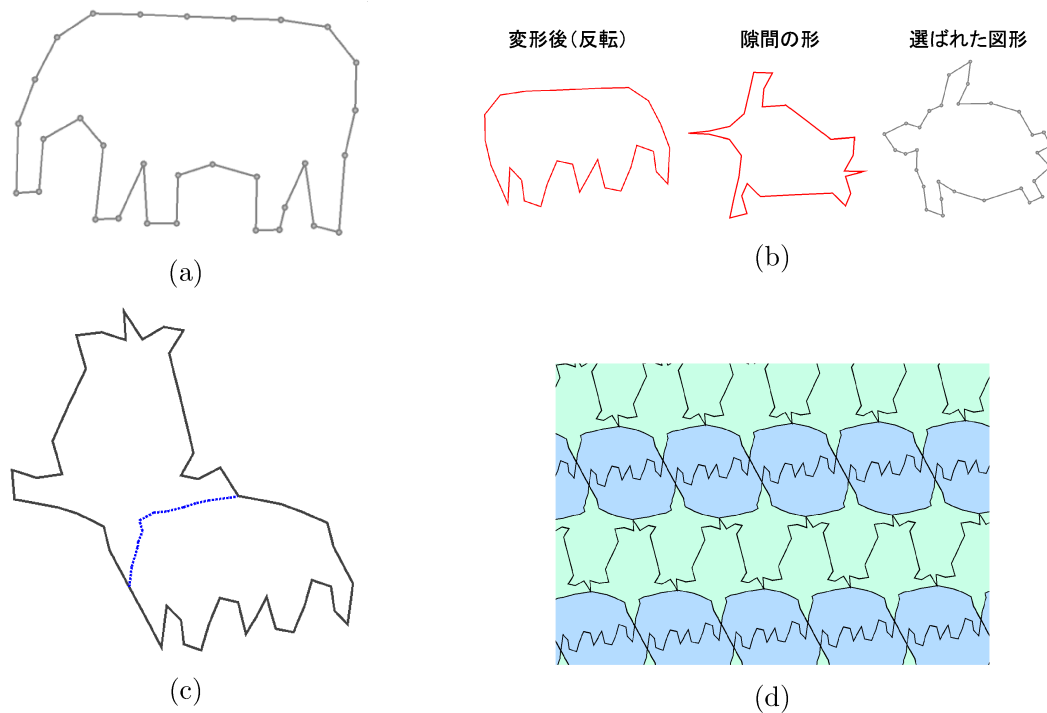


図 26: ゾウの実験結果

図 25 は、エッシャーの作品「空と水 I」(1938)[1],[2] を模倣したものであるが、一方の図形を入力として与えると、もう一方の図形として正しい図形が選ばれることが確認できた。図 24 と図 26 から提案手法の有効性が確認できるが、特に図 26 の (b) と (c) を比較すると、4.1 節の接合法や酒井の手法によりタイリングを作り直したことで、データベースから選ばれたカメの図形の変形が小さくなったことが分かる。また、表 2 より、図形の選択に 2 分から 3 分ほど時間がかかり、タイリング生成には時間はほとんどかかっていないことが分かる。

6 まとめ

本稿では、エッシャー風タイリング問題の中でも、特に 2 種類の図形を扱う問題に対して、手法の提案を行った。まず、2 つの図形を入力し、それらを接合することにより 1 つの図形とみなし、それに対して小泉らの手法や酒井の手法を適用してタイリングを生成する方法を提案した。その実験結果より、2 つの入力図形の組み合わせが解の質に大きく影響を与えることを示した。そこで次に、入力図形を 1 つのみとし、もう一方の図形をあらかじめ用意しておいたデータベースの中から選び出すことを考えた。図形の選択方法や、図形が選ばれた後のタイリング生成方法に関して提案を行い、数値実験によりその有効性を示した。

今後の課題としては、まず計算時間の短縮を図ることが挙げられる。これは、5.3 節の実験では 28 種類考慮すべきタイリングタイプを 3 つのみに絞って行ったことを考えると、全てのタイプで実行するにはおそらく 15 分から 30 分程かかると予想され、実用的な時間とは言えないためである。また、提案手法では、isohedral タイリングのタイルを、タイリング頂点を結ぶ線で分割した

が、実際は辺の途中で分割してもよいはずであり、表現できるタイリングが制限されてしまっているという問題もある。ただし、図形の選択の段階でそこまで考えると、上で述べた計算時間がさらに増えることになり、好ましくない。そこで、図形の選択段階は今のままとし、タイリング生成段階で、図形を接合する部分にある程度幅を持たせたり、タイリング頂点の位置をずらしたりすることで、ある程度柔軟に表現できると考えられる。そのようにして、解の質の向上を図ることも今後の課題である。

参考文献

- [1] M. C. Escher: M. C. エッシャー グラフィック, タッシェン・ジャパン, 2008.
- [2] M. C. Escher - The Official Website, <http://www.mcescher.com/>.
- [3] B. Grünbaum, G. C. Shephard: Tiling and Patterns, W. H. Freeman, 1987.
- [4] C. S. Kaplan: Introductory Tiling Theory for Computer Graphics, Morgan & Claypool Publishers, 2009.
- [5] C. S. Kaplan, D. H. Salesin: Escherization, Proceedings of SIGGRAPH, pp. 499-510, 2000.
- [6] C. S. Kaplan, D. H. Salesin: Dihedral Escherization, Proceedings of Graphics Interface, pp. 255-262, 2004.
- [7] 川出 静: 2種類の図形によるタイリング生成ー図形の接合を用いたアルゴリズムの構築ー, 名古屋大学工学部物理工学科 卒業論文, 2013.
- [8] D. G. Kendall: Shape Manifolds, Procrustean Metrics, and Complex Projective Spaces, Bulletin of the London, Mathematical Society, Vol. 16, pp. 81-121, 1984.
- [9] 小泉 拓: エッシャー風タイリングの自動生成, 東京大学大学院情報理工学系研究科数理情報学専攻 修士論文, 2010.
- [10] H. Koizumi, K. Sugihara: Maximum Eigenvalue Problem for Escherization, Graphs and Combinatorics, Vol. 27, pp. 431-439, 2011.
- [11] 酒井 翔平: エッシャー風タイリング問題の数理モデルについてー制約条件の緩和及びその最適化手法ー, 名古屋大学大学院工学研究科計算理工学専攻 修士論文, 2013.
- [12] 杉原 正顕, 室田 一雄: 線形計算の数理, 岩波書店, 2009.