

Title	正多面体とその拡張クラスに対する折り判定問題の研究
Author(s)	鎌田, 斗南
Citation	
Issue Date	2021-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/17090
Rights	
Description	Supervisor:上原 隆平, 先端科学技術研究科, 修士(情報科学)

修士論文

正多面体とその拡張クラスに対する折り判定問題の研究

鎌田斗南

主指導教員 上原隆平教授

北陸先端科学技術大学院大学
先端科学技術研究科
(情報科学)

令和3年3月

Abstract

Regular polyhedra are a class of convex polyhedra that have congruent regular polygons as faces and congruent apices (as vertices). Regular polyhedra consist of regular tetrahedron, regular hexahedron, regular octahedron, regular icosahedron, and regular dodecahedron. Regular polyhedra have a long history going back to the book “Elements” written by Euclid. Since then, it has been researched with related to the group theory, the theory of the algebraic equation, and so on.

Polyhedron is solid in 3D, however, its surface consists of connected figures in 2D. There is a way of representation focused on the surface of the polyhedron called “unfolding”. A history of unfolding is back to a book written by Dürer in 1525. In this book, polyhedra are represented by the placement of faces according to their adjacency. Since they can be regarded as polygons obtained by cutting edges and open, they are called edge unfolding. An edge unfolding is still an active topic. For example, there is an open problem “whether every convex polyhedron has a non-overlapping edge unfolding”. On the other hand, when any place other than edges is admitted to be cut, it is called (general) unfolding. Against the number of edge unfolding is finite for a polyhedron, the number of general unfolding is (uncountable) infinite. Therefore, the relationship between a polyhedron and its unfolding is difficult. Even simple polyhedron like regular polyhedron, there are only a few results except for regular tetrahedron by Akiyama.

In computational geometry, a problem called “folding problem” has been researched. The folding problem asks if a polygon P is unfolding of a polyhedron Q for given P and Q . First, a polynomial-time algorithm was developed in the case that P is a polyomino and Q is a box of integer size. After that, it is extended to a general polygon P .

In this research, we recapture the folding problem as a characterization of an unfolding and organize the previous framework. As a result, we develop polynomial-time algorithms for a case that Q is a regular polyhedron which is extended to a deltahedron, a tetramonohedron, and a box of integer size. A part of this research was published at the 32nd Canadian Conference on Computational Geometry.

目次

第1章	はじめに	1
第2章	準備	2
2.1	基本的な定義と性質	2
2.2	折りと展開についての定義と諸性質	4
2.3	多面体のクラス	5
第3章	結果	7
3.1	アルゴリズムのフレームワーク	7
3.2	初期配置候補の列挙	8
3.3	スタンプング	10
3.4	貼り合わせ確認	14
3.5	Qが等面四面体の場合の対応点	16
3.6	対応点の位置の列挙	21
3.7	まとめ	25
第4章	Future Work	26
第5章	謝辞	27

目 次

2.1	多角形の例	3
2.2	多角形の内角, 両隣点	3
2.3	多面体	3
2.4	曲率	4
2.5	切断木とそれによる展開図	4
2.6	整直方体の例	6
2.7	高次デルタ多面体の例	6
2.8	正十二面体	6
2.9	等面四面体の例	6
3.1	3つの場合	11
3.2	(1)	12
3.3	(2)	13
3.4	(3)	13
3.5	長さ1の線分上のスタンプ	14
3.6	葉が取りうる3つの状態	16
3.7	$e_v, r_v, \alpha_v^+, \alpha_v^-, g_v$ の定義	18
3.8	v_1 :分割頂点, v_2 :等分頂点, $v_3:v_2$ に従属する準等分頂点	18
3.9	準等分頂点の対応点	19
3.10	等面四面体の切断木の分類, 破線はパスの略記	20
3.11	$\angle(e_{v_1}, e_{v_2}) = \pi$ かつ $\angle(e_{v_3}, e_{v_4}) = \pi$ の場合	21
3.12	対応点間のベクトル	24

第1章 はじめに

正多面体とは、全ての面が単一の正多角形から構成され、頂点が全て合同な凸多面体のクラスであり、正四面体、正六面体、正八面体、正十二面体、正二十面体の5つからなる。正多面体の歴史は古く、ユークリッドの原論の時代からその存在が知られている。以降、代数方程式や群論といった数学的な対象の誕生と密接な関わりを持ちながら研究が行われてきた。

多面体は3次元空間内の立体であると同時に、その表面はつなぎ合わされた2次元図形で構成されている。このような表面の構造に注目して多面体を表現するための方法の一つに、「展開図」と呼ばれる多角形を用いたものがある。展開図の歴史は1525年のDürerによる著書「Underweysung der messung, mit den zirckel un richtscheyt, in Linien ebenen unnd gantzen corporen」に遡る。多面体はこの中で、隣接状況に従って各面を平面上に並べてできる多角形で表現されている。これは辺に沿って多面体を切り開いて得られる多角形とみなせるため、辺展開図と呼ばれる。辺展開図に関しては「全ての凸多面体に重なりのない辺展開図が存在するか」という未解決問題が知られており、現在も重要な研究対象である。また、切り開く箇所を辺に限定しないものは（一般）展開図と呼ばれる。一つの多面体に対して、辺展開図の個数が有限個である一方で、展開図は非可算無限個存在する。そのため多面体とその展開図の関係を考えることは難しく、正多面体のように単純な多面体であっても、Akiyama等の正四面体の幾何学的な特徴付けの結果を除いて、知られていることはほとんどない。

他方、計算幾何学においては「折り判定問題」と呼ばれる問題が研究されている。折り判定問題とは、与えられた多角形 P と多面体 Q に対して、 P が Q の展開図かどうか、つまり P を折って Q を得ることができるかを判定する問題である。この問題は最初に、 Q が辺長が整数の直方体、 P がポリオミノ（正方形の辺同士を貼り合わせて得られる多角形）の場合に対する多項式時間アルゴリズムが提案され、後に P が一般の多角形の場合に拡張された。

本研究では、折り判定問題を展開図の特徴づけの問題として捉え直すことで、多面体のクラスに応じた展開図の特徴を解析し、既存の折り判定アルゴリズムのフレームワークを整理した。これによって、アルゴリズムの適用範囲を Q が正多面体または、その一般化であるデルタ多面体、等面四面体、整直方体の場合まで拡張した。この結果の一部はThe 32nd Canadian Conference on Computational Geometryにて発表された。

第2章 準備

本章では、多角形と多面体、それらの間の操作である折りと展開についての定義とその性質について述べる。

2.1 基本的な定義と性質

定義 1. (多角形 P , 辺集合 $E(P)$, 頂点集合 $V(P)$, 周長 L , 周 ∂P)

頂点数 n の多角形 P とは, \mathbb{R}^2 の座標の異なる n 個の点の集合 $V(P)$ と, $V(P)$ の要素を並べた配列 $Array_P = (p_0, p_1, \dots, p_{n-1})$ によって定まる \mathbb{R}^2 上の閉じた一次元図形である (図 2.1).

多角形 P に対して, $V(P)$ の元を頂点と呼ぶ. $Array_P$ 上で連続する 2 つの頂点を端点にもつ線分を P の辺と呼び, 辺の集合を $E(P)$ で表す. また, 辺上の点 (端点を含む) の集合を周と呼び, ∂P で表す. 辺長の合計を周長とよび, L で表す.

$p \in \partial P$ に対して, P の周上で反時計回りで最も近い頂点を p の次頂点とよび $ccw(p)$ で表す. 時計回りで最も近い頂点を p の前頂点と呼び $cw(p)$ で表す. $ccw(p), cw(p)$ を合わせて両隣点と呼ぶ. p の内角 $\alpha(p)$ を $\angle(cw(p), p, ccw(p))$ で定める (図 2.2).

注意 2. (内角 π の頂点)

P の頂点 p が $\alpha(p) = \pi$ を満たす時, p とその両隣点は同一直線上にある. よって p は辺上の点とみなすことができる. つまり, $V(P)$ から内角 π の頂点を取り除いたり, 辺上の点を内角 π の頂点として $V(P)$ に追加しても P の形状は変化しない. 従って, $Input$ として与えられる多角形は, 内角が π の頂点は全て取り除かれているものとする. また, アルゴリズムの中で適宜, 辺上の点を頂点集合に追加することがある.

注意 3. (多角形の単純性)

多角形を定義する時, 「すべての辺が互いに交差しない」という条件を課す場合がある. これを満たす多角形は単純多角形と呼ばれる. しかし, 定義 11 で定める「展開図」は一般に単純でない, 従って, 本研究では多角形の定義に単純性は課さない.

定義 4. (凸多面体 Q , 辺集合 $E(Q)$, 頂点集合 $V(Q)$, 面集合 $F(Q)$)

三次元空間内の半空間の共通部分 Q は凸多面体を形成する. これを面をなす

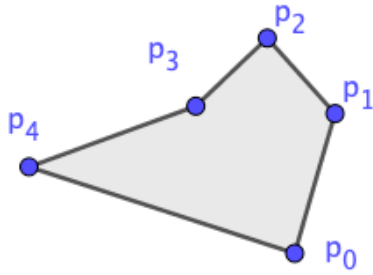


図 2.1: 多角形の例

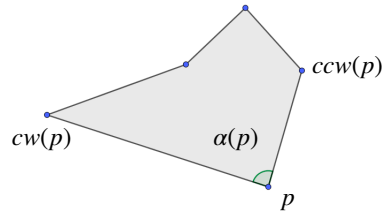


図 2.2: 多角形の内角, 両隣点

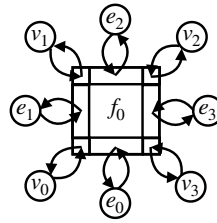
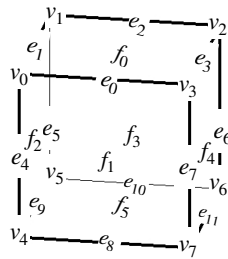


図 2.3: 多面体

ラベル付き多角形の集合 $F(Q) = \{f_0, f_1, \dots, f_{m_f-1}\}$ と, 辺と頂点のラベルの集合 $E(Q) = \{e_0, e_1, \dots, e_{m_e-1}\}$, $V(Q) = \{v_0, v_1, \dots, v_{m_v-1}\}$ を使い, $F(Q)$ の要素のそれぞれの辺と頂点と, $E(Q)$ と $V(Q)$ の間の対応関係を定めたリスト構造 $List_Q$ で表す (図 2.3).

面上の点の集合と辺上の点の集合と頂点集合の和集合を表面と呼び, $S(Q)$ で表す.

注意 5. 以降, 凸多面体を単に多面体と呼ぶ.

多面体 Q の表面上の点は, 面の内部に含まれるか, 辺に含まれるか, 頂点に一致する. そこで, それぞれの面 f や辺 e には, 独立した座標系 $(x, y)_f$, $(x)_e$ が定められているものとする. これを用いて Q の表面上の点の座標を次のように定める.

定義 6. ($p \in S(Q)$ の座標 $Coord_Q(p)$)

Q の表面上の点 p に, 座標 $Coord_Q(p)$ を次のように定める.

- ・ p がある面 $f \in F(Q)$ の内部に含まれる場合, $Coord_Q(p) := (x, y)_f$
- ・ p がある辺 $e \in E(Q)$ の内部に含まれる場合, $Coord_Q(p) := (x)_e$
- ・ p がある頂点 $v \in V(Q)$ に一致する場合, $Coord_Q(p) := v$

定義 7. ($v \in V(Q)$ の曲率 $\kappa(v)$, 余曲率 $\sigma(v)$)

多面体 Q の頂点 v に対して, v の余曲率 $\sigma(v)$ を v に対応する頂点を持つ面の内角の総和で定める. また v の曲率を $\kappa(v) := 2\pi - \sigma(v)$ で定める (図 2.4).

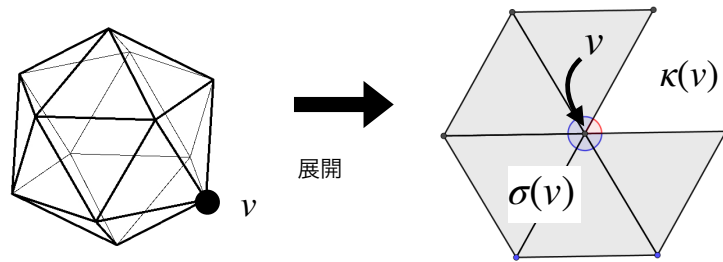


図 2.4: 曲率

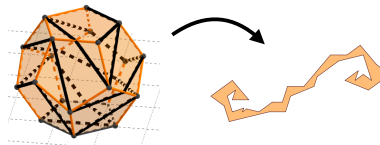


図 2.5: 切断木とそれによる展開図

注意 8. (凸多面体の曲率)

任意の凸多面体 Q に対して, $\forall v \in V(Q), 0 < \kappa(v) < 2\pi$ が成り立つ.

定理 9. (デカルトの定理, 離散ガウスボンネの定理)[9]

任意の凸多面体 Q に対して, $\sum_{v \in V(Q)} \kappa(v) = 4\pi$

2.2 折りと展開についての定義と諸性質

定義 10. (切断木 CT)

多面体 Q の表面上の線分集合 CT が切断木であるとは, CT が木構造をなし, Q の全ての頂点が CT のいずれかの線分に含まれることである.

定義 11. (多面体 Q の CT による展開)

多面体 Q とその切断木 CT に対して, Q の各面を CT によって分割し, 各辺が平坦になるように回転操作を施し, 平面上の多角形を得ることを, Q を CT で展開するという. また得られた多角形 P を Q の CT による展開図という (図 2.5).

注意 12. 多面体 Q が凸である時, 展開図として連結な平面多角形を得るためには, それを与える Q の表面上の線分集合が, Q の全ての頂点を含み, 木構造をなすことが必要である [9]. 従って, 切断木による展開のみを考えれば良い.

注意 13. 凸多面体の展開を考える際, 多面体の頂点以外の箇所にある切断木の次数 1 の頂点は展開図の形状に変化を与えない [12]. 従って切断木は多面体の頂点だけに葉を持つものとする.

多角形 P を多面体 Q の切断木 CT による展開図とする。展開の操作を逆再生して考えると、これは P によって Q の表面を覆う操作とみなすことができる。この時、 P の周はお互いに貼り合わされることで CT を形成している。このように P に折り目を入れて Q を得る操作を折りと呼ぶ。 P を Q に折れるかどうかは、 P を Q の表面に貼り付けることで P の周が重なりや隙間なく張り合わされるかどうかで帰着できる。 CT の辺が Q の辺を直線によって横切る場合、 P の一つの辺は Q の複数の面に対応する。そこで、 P の Q への折りを P の辺を適当に細分した多角形と、その各辺の多面体の表面上への対応関係の組によって以下のように定義する。

定義 14. (多角形 P の多面体 Q への貼り付け方 γ)

与えられた多角形 P と多面体 Q に対して、いくつかの P の辺上の点を $V(P)$ に加えて得られる多角形 P_γ と、写像 $\phi_\gamma : \partial P \rightarrow S(Q)$ の組 $\gamma = (P_\gamma, \phi_\gamma)$ が次を満たす時、 γ を P から Q への貼り付け方と呼ぶ。

- (1) 任意の $e \in E(P_\gamma)$ に対して、 $\phi_\gamma(e)$ は Q の一つの面に含まれ、 $|e| = |\phi_\gamma(e)|$
- (2) 任意の頂点 v に対して、 v に入射する二辺を $e, e' \in E(P_\gamma)$ とすると、 $\phi_\gamma(e)$ と $\phi_\gamma(e')$ は $\phi_\gamma(v)$ を共有し、 $\angle(e, v, e') = \angle(\phi_\gamma(e), \phi_\gamma(v), \phi_\gamma(e'))$

注意 15. (貼りつけ方の構成)

P_γ と ϕ_γ の $V(P_\gamma) \subset \partial P$ 上の値を定めると、 ϕ_γ の ∂P 全体上での値が一意的に決定する。

定理 16. (折りと展開の関係)

多角形 P の面積と多面体 Q の表面積が等しい時、次は同値である。

- (1) P が Q の展開図である
 - (2) $\phi_\gamma(\partial P)$ が Q の表面上の全域木となる貼り付け方 γ が存在する。
- またこれが成り立つ時、 P は Q に γ で折れるという。

Proof. ((1) \Rightarrow (2)) 展開図 P を与える Q の切断木を CT とする。この時、 P の周上の各点の CT 上の点への対応関係に注目すると、これは写像を成す。この写像を ϕ_γ とし、 P の周上の点の中で ϕ_γ の値が Q の辺上の点であるものを全て $V(P)$ に追加することで得られる多角形を P_γ とすると、これは P から Q への貼り付け方である、

((2) \Rightarrow (1)) $\phi_\gamma(\partial P)$ は Q の表面上の全域木であるため、これを切断木とみなすことができる。この切断木によって得られる Q の展開図は P に一致する。 \square

2.3 多面体のクラス

本節では本研究で扱う多面体のクラスについて述べる。

(1) それぞれの辺が整数長の直方体を整直方体と呼び、その集合を IntBox を表す(図 2.6)。

(2) 各面が単位三角形あるいは単位三角形を接着した多角形を面にもつ凸多面体を高次デルタ多面体と呼び, その集合を HiDelta で表す (図 2.7).

(3) 12 個の単位正五角形を面を持つ凸多面体を正十二面体とよび, その (一元) 集合を Dodeca で表す (図 2.8).

(4) 4 つの合同な正三角形を面を持つ多面体を等面四面体とよび, その集合を Tetramono で表す. この時, 面をなす三角形の最短の辺長を 1 とする (図 2.9).

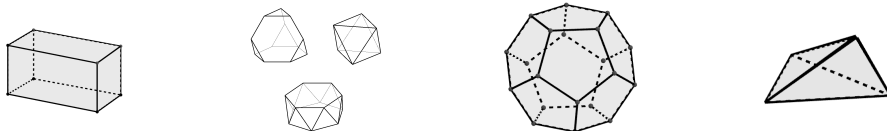


図 2.6: 整直方体の例

図 2.7: 高次デルタ多面体の例

図 2.8: 正十二面体

図 2.9: 等面四面体の例

第3章 結果

折り判定問題を次のように定める.

Input : 凸多面体 Q を表す $List_Q$, 多角形 P を表す $Array_P$
(一般性を失うことなく, P の面積と Q の表面積は一致すると仮定する)

Output: P を Q に折れる全ての貼り付け方

本研究の主結果は次の定理を示したことにある.

定理 17. (主結果)

$Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Tetramono}$ の場合, 折り判定問題を $O(n^2(L+n)^2)$ 時間で解くアルゴリズムが存在する. また, $Q \in \text{Dodeca}$ の場合, 折り判定問題を $O(n^2(L+n)^4)$ 時間で解くアルゴリズムが存在する.

注意 18. 計算モデルは *Real-RAM* モデルを仮定する. つまり任意の実数同士の演算は誤差なしに定数時間で実行できるとし, この演算の回数によって計算時間を評価する.

この結果は, 先行研究 [6],[7] の結果を改善し, 対象となる多面体の範囲を拡大したものである. また $Q \in \text{Tetramono}$ の場合の結果は [3],[4] で示された幾何学的な特徴づけのアルゴリズム化に当たる. 先行研究との比較についての詳細は 3.7 章で述べる.

3.1 アルゴリズムのフレームワーク

本節では, 折り判定アルゴリズムのフレームワークを述べる.

定理 16 より, 与えられた多角形 P が多面体 Q の展開図かどうかという問題は, P から Q への貼り付け方 γ で $\phi_\gamma(\partial P)$ が Q の表面上の全域木をなすものが存在するかどうかという問題に帰着できる. そこで, 本研究では可能性のある貼り付け方全てを列挙して, それらが Q の表面上の全域木を与えるかを確認していく. 貼り付け方は, 空間内に配置された P と Q に対してスタンピングと呼ばれる操作を行うことで得られる. スタンピングから得られる貼り付け方は, P と Q の配置か

ら一意的に決定する．考える必要がある配置を「初期配置候補」と呼び，この列挙について3.2節で述べる．それぞれの初期配置候補に対するスタンプングの操作について3.3節で述べる．スタンプングによって得られた張り合わせ方が全域木を与えるかどうかの判定を行う方法について3.4節で述べる．

3.2 初期配置候補の列挙

この節では，初期配置候補を列挙する方法について述べる．初期配置を考える上で重要なのは多面体と多角形の相対的な位置関係である．つまり，多面体の配置を一つ固定し，多角形の配置についてのみ列挙を考えれば良い．そこで多面体の初期配置を次のように定める．

定義 19. (多面体の初期配置)

多面体 Q に対して，一つの面 f と f の頂点 v を選んで， e を v に時計回りに入射する辺とするととき， v が原点に一致し， f が xy 平面に， e が x 軸に含まれるような Q の配置を， Q の初期配置 (v, e, f) と呼ぶ．

Q の初期配置を (v, e, f) とし，多角形 P の初期配置について考える． P の配置は周上の2つの点を選び，その座標を決定することで定めることができる．ここで次の概念を導入する．

定義 20. ($v \in V(Q)$ の対応点 $CP(v)$)

多角形 P が多面体 Q に貼り付け方 γ で折れるとする． $v \in V(Q)$ に対して， $CP(v) := \{p \in \partial P : v = \phi_\gamma(p)\}$ を v の対応点の集合と呼ぶ．

注意 21. $CP(v)$ の要素数は， $\phi_\gamma(\partial P)$ を切断木とみなした時の v における次数に等しく， $CP(v)$ の各要素の内角の総和は $\sigma(v)$ に等しい．

ある v, e, f を選び， Q の初期配置を (v, e, f) とした場合の v の対応点について考える． Q の初期配置を (v, e, f) としたことから， q は原点に一致している．つまり， ∂P の点で $CP(v)$ に含まれる点 p を探索することができれば， p が原点に一致する P の配置を考えることができる．ここで次の定理が成り立つ．

定理 22. (対応点の基本定理)

$v \in V(Q)$ が $\kappa(v) \neq \pi$ を満たせば， $CP(v)$ は $V(P)$ の点を一つ以上含む．

Proof. $v \in V(Q)$ の対応点の中に内角が π ではない頂点が一つ以上存在することを確かめれば良い．注意 21 より $CP(v)$ に含まれる点の内角の和は $\sigma(v)$ に等しい．また，注意 8 より $0 < \sigma(v) < 2\pi$ である．従って， $CP(v)$ に含まれる内角 π の点の個数は高々1つである． $CP(v)$ が内角 π の点を一つだけ含む場合，それを p とすると $\kappa(v) \neq \pi$ より， $CP(v)$ は p 以外の頂点を一つ以上含む．よって，内角 π でない対応点が一つ以上存在する． \square

以上より、 Q が曲率が π でない頂点を持てば、その対応点の中に P の頂点が一つ以上含まれる。これを v として選べば、 p は $V(P)$ の中から選ぶことができる。そこで、 Q が曲率が π でない頂点を持つ条件について考える。

定理 23. (等面四面体の特徴づけ)

多面体 Q について、次は同値

(1) Q が等面四面体である

(2) $\forall v \in V(Q), \kappa(v) = \pi$

Proof. (1) \Rightarrow (2)

Q を等面四面体とし、面をなす三角形を T とする。また T の各頂点を A, B, C とする。 Q の頂点 v を任意に選ぶ。 v に集まる 3 つの面の頂点について考える。 T を 4 枚接着して Q を作る際に接着される辺の長さを一致させる必要があることを考慮すると、 v には T の頂点 A, B, C が一つずつ集まっている。 A, B, C の内角の和は π であるため、 $\kappa(v) = \pi$ が成り立つ。

(2) \Rightarrow (1)

Q の頂点を v_1, v_2, v_3, v_4 とし、 Q を線分 v_1v_2, v_1v_3, v_1v_4 で切り開くことで得られる多角形 P を考える。 P の周上での v_2, v_3, v_4 の対応点を考えると、 Q の各頂点の曲率が π であることから、それらの内角は全て π である。従って P は v_1 の 3 つの対応点を頂点にもつ三角形である。また、 P は Q を辺に沿って展開して得られたため、 v_2, v_3, v_4 の対応点はそれぞれの辺の midpoint に位置する。従って P は線分 v_2v_3, v_2v_4, v_3v_4 によって合同な 3 つの三角形に分割され、それらが Q の面に対応する。従ってこれは等面四面体である。□

定理 24. Q が等面四面体でないならば、 Q は曲率 π でない頂点を 2 つ以上もつ

Proof. Q が等面四面体でないことから、曲率 π でない頂点が一つ以上存在する。よって示すべきなのは曲率 π でない頂点がさらにもう一つ存在することである。もし Q が曲率 π でない頂点 v を一つだけ持つならば、残りの頂点の曲率の和は π の整数倍である。しかし、これに v を加えた Q のすべての頂点の曲率の合計が π の整数倍でないことになり、定理 9 に反する。□

以上より、 Q が等面四面体でない時には、曲率 π でない頂点が 2 つ以上存在することがわかる。つまり、 Q が等面四面体でない場合、 q として曲率 π でない頂点を取ることができ、 q のある対応点 p は $V(P)$ に含まれる。つまり、 $V(P)$ のそれぞれの要素に対して、それが p であると仮定し、 p が原点に一致する P の配置について考えれば良い。 P の配置を決定するためには、 P の周上の点をもう一つ選んで、その座標を決定する必要がある。定理 24 は v の他にもう一つの曲率 π でない頂点が存在することを保証している。これを v' とすると、 v' のある対応点 p' も同様に $V(P)$ に含まれる。よって $V(P) - \{p\}$ の元それぞれに対して、それが p' と仮定した場合を考えれば良い。最後に、 p' の座標 $p' = (p'_x, p'_y)$ を決定すれば P の配

置が決定する．頂点間の距離を考えると p'_x, p'_y は $p_x'^2 + p_y'^2 = |p, p'|^2$ を満たす必要がある． p は原点に位置するため， p' は，平面上の原点中心，半径 $\sqrt{p_x'^2 + p_y'^2}$ の円 C 上に含まれる必要がある． C の円周上には無数に点があるが，これら全てを p' の座標として考える必要はない．可能性のある p' の位置は Q の形状に依った方法で列挙することができ，候補の個数は多角形の頂点数と周長の多項式で抑えることができる．この方法については 3.6 節で別途考察する．また， Q が等面四面体の場合には曲率 π でない頂点は存在しないため，対応点を多角形の辺上の点の中から探すために別の方法を用いる必要がある．これについては 3.5 節で考察する．以降では以下の形で定められる何らかの初期配置が与えられたものとして考える．

定義 25. (多角形 P の初期配置)

多角形 P の初期配置は，周上の 2 点 $p, p' \in \partial P$ を選んでその座標 ($p = (0, 0), p' = (p'_x, p'_y)$) で定める．

つまり， P と Q が $Array_P, List_Q$ として与えられた時，それらの初期位置候補の列挙は次の手順で行うことができる．まず P の周上で Q の頂点の対応点になりうる二点 p, p' の組合せを列挙する．これは Q が等面四面体である場合には 3.5 節の Algorithm 3，その他の場合は $V(P)$ から 2 つを選んだ全ての組合せで与えられる．また， Q が等面四面体である場合は p, p' を (内角 π の頂点として) $Array_P$ に追加する．次に，それらに対して 3.6 節で与えられる Algorithm 4 を用いて p, p' の座標 ($p = (0, 0), p' = (p'_x, p'_y)$) を決定し， $Array_P$ の各要素を ($p = (0, 0), p' = (p'_x, p'_y)$) に従って座標変換する．最後に， $Array_P$ を p が先頭になる様に並べかえる (これらは $O(n)$ 時間で実行できる)．

以後，多面体の各頂点に対して操作を行うことがあるが，これは以下の定理より線形時間で行うことができる．

定理 26. Q の頂点数は $O(n)$

Proof. 定理 22 より，多面体の曲率が π でない頂点の個数は P の頂点数 n 以下より少ない．また，定理 9 より曲率 π の頂点は高々 4 個である．従って Q の頂点数は $n + 4$ 以下．

□

3.3 スタンピング

この節では，初期配置の候補が一つ与えられた時，そこから張り合わせ方を得る方法について述べる．前節の議論により， P の周上の点 p, p' とその座標 ($p = (0, 0), p' = (p'_x, p'_y)$) が定められ， p が先頭かつ ($p = (0, 0), p' = (p'_x, p'_y)$) に従って座標変換された P の頂点の座標を並べた配列 $Array_P$ が得られている．また $v \in V(Q), e \in E(Q), f \in F(Q)$ によって Q の初期配置 (v, e, f) が定められている

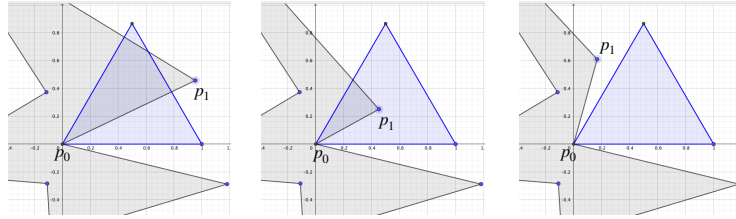


図 3.1: 3つの場合

(Q が等面四面体でない時, v は $\kappa(v) \neq \pi$ を満たす様に選ばれている). ここからスタンプと呼ばれる手法で一つの貼りつけ方を得る. 貼りつけ方は, 注意 15 より, $V(P)$ に周上の点をいくつか追加した $V(P_\gamma)$ の各要素の Q の表面上の座標によって定めることができる. そこで, 配列 $Array_\Gamma$ を用意し, そこに Q の表面上の座標を格納していく.

スタンプングとは, P の上で Q を仮想的に転がしていくことで P の周上の点と Q の表面上の点の対応関係を定める技法である. これは [3][4] で用いられた技法を改良している.

まずはアルゴリズムの概要を述べる. Q を「転がす」操作は, Q の底面の辺を一つ選んで, その辺を共有する Q の面を底面に隣接するように描画する操作の繰り返しであると考えられる. そこで, 底面を管理するための Q の面へのポイント $base$ を用意し, f に初期化する. Q の位置が定められているため, $base(=f)$ にはある配置が定められている. これに従って平面上に $base$ が指す面を平面上に描画し, それを P_{base} とする. ここから, P の周を反時計回りに辿りながら, P の周と Q の底面の辺が交点を持つたびに $base$ を更新し, 現在の底面の配置に従って平面上の多角形 P_{base} を描画する. これを繰り返すことで P の周上の点に Q の表面上の点を対応させていく. P_{base} として描画される平面上の多角形をスタンプと呼び, その集合を $STAMP$ で表す.

アルゴリズムをより詳細に述べる. まず, $V(P)$ の先頭の要素 p_0 は v に対応するため, $\phi_\gamma(p_0) = Coord(v)$ である. これを $Array_\Gamma$ に一つ目の要素として収める.

次に, P_{base} の周と P の辺 (p_0, p_1) の関係について考えると, (1) (p_0, p_1) は P_{base} の周と交点を持つ, (2) (p_0, p_1) は P_{base} の周と交点を持たず, p_1 が P_{base} の内部にある, (3) (p_0, p_1) は P_{base} の周と交点を持たず, p_1 が P_{base} の外部にある, のどれかが成り立つ. それぞれの場合について, (p_0, p_1) 上の点の ϕ_γ の値について考える (図 3.1).

(1) の場合 (図 3.2), 交点を c とすると, p_0 から c まで線分 $(p_0, c) \subset \partial P$ は Q の面 $base$ 上の線分に対応し, c は P_{base} の頂点か辺に含まれる. また, $(c, p_1) \subset \partial P$ 上の点は $base$ とは異なる面に対応している. このことに注意してそれぞれの点が Q にどの様に対応するかを考える. c は Q の頂点または辺上のある点 c_Q に対応するので $\phi_\gamma(c) = Coord_Q(c_Q)$ と定めることができる. これを $Array_\Gamma$ の最後尾に追加する. 次に, (c, p_1) 上の点が Q の表面上にどのように対応するかを考える. c が

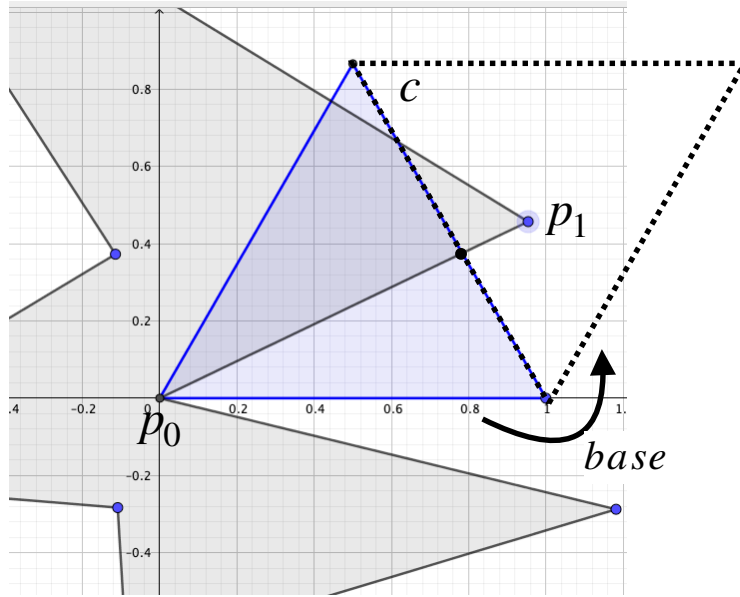


図 3.2: (1)

P_{base} の辺に含まれる場合, その辺に対応する Q の辺を e' とすると, (c, p_1) 上の点は, e' を $base$ と共有する面 f' に対応する. よって $base$ を f' に更新し, すでに位置が定まっている e' に従って $base$ を平面上に描画し, P_{base} とする. 改めて (c, p_1) と P_{base} の周の関係を考えると (1),(2),(3) と同様の場合分けに帰着できる. c が P_{base} の頂点に含まれる場合, その頂点に時計回りに入射する P_{base} の辺に対応する Q の辺を e' とする. これを c が P_{base} の辺に含まれる場合と同様に, (c, p_1) と P_{base} の周の関係を考える事で (1),(2),(3) の場合分けに帰着できる.

(2) の場合 (図 3.3), p_1 は $base$ の内部の点 v に対応している. よって, $\phi_\gamma(p_1) = \text{Coord}_Q(v)$ と定めることができる. これを Array_Γ の最後尾に追加する. ここで (p_1, p_2) と P_{base} の周の関係を考えると (1),(2),(3) と同様の場合分けに帰着できる.

(3) の場合 (図 3.4), $base$ を $base$ と e を共有する面に更新し, e の位置に従って P_{base} を定める. これを (1),(2) を満たすまで繰り返す事で, (1),(2) に帰着できる.

以上の操作を P の周を一周するまで繰り返すことで, P の周上の点に Q の表面上の座標を取めた配列 Array_Γ が得られる. Array_Γ の先頭と最後尾は共に p の Q の表面上の座標が収納されている. これらが一致しない場合, それは明らかに正しい貼り付け方を与えない. 従ってこのような場合, $false$ を出力する. また, 処理の途中で $p \in \partial P$ に割り当てた $\phi_\gamma(p)$ が $\alpha(p) = \sigma(\phi_\gamma(p_0))$ を満たせば, p は切断木における葉に対応する. 後の処理に用いるため, 配列 $\text{Array}_{\text{Leave}}$ を用意して, この様な頂点へのポインタを格納しておく. さらに, Q の各頂点に対応する配列 Array_Q を用意して, その頂点に多角形の周上の点が割り当てられるたびにチェックを入れる. チェックが入らない要素が存在する時, その頂点は切断木に含まれないため, γ は全域木を与えない. この様な場合には $false$ を出力する. 詳細は

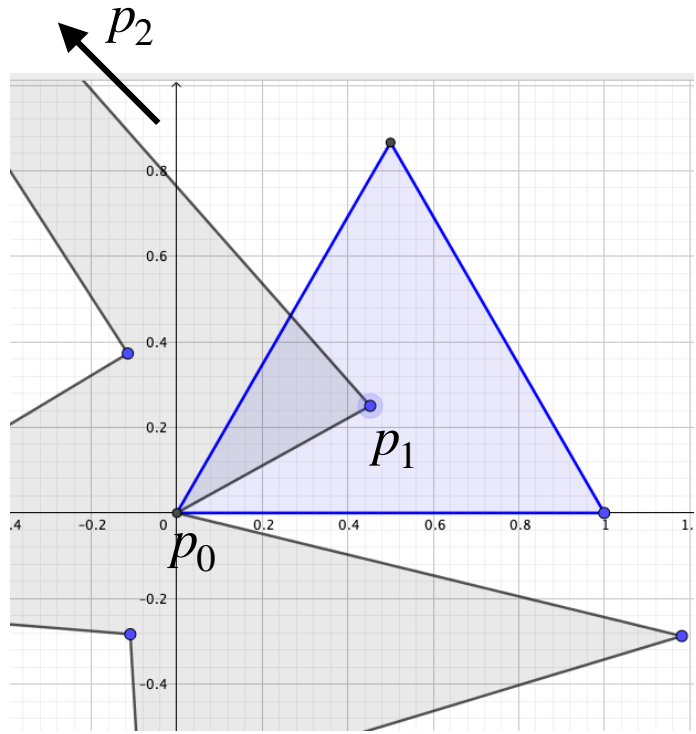


图 3.3: (2)

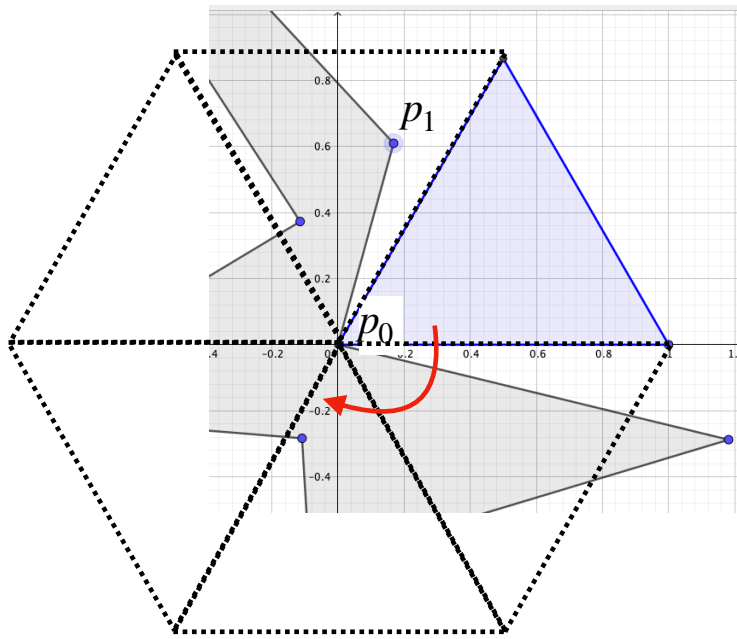


图 3.4: (3)

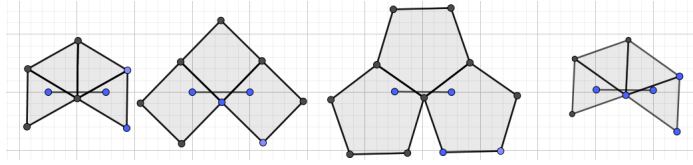


図 3.5: 長さ 1 の線分上のスタンプ

Algorithm 1 に示す.

最後に, スタンプの処理の計算時間を考える. 次が成り立つ.

定理 27. (スタンプの計算時間)

多角形 P と多面体 $Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Dodeca} \cup \text{Tetramono}$ に対してスタンプングを行って得られる Array_Γ の要素数は $O(L+n)$ である. また, この際の STAMP の要素数は $O(L+n)$, *Algorithm 1* の計算時間は $O(L+n)$ 時間である.

Proof. Array_Γ の要素数は (1) の場合と (2) の場合が発生する回数の和に等しい. P の一つの辺 e を選んで, その上で (1),(2) の場合が発生する回数に注目する. この回数は e を分割するスタンプの個数に等しい. $Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Dodeca} \cup \text{Tetramono}$ のとき, スタンプは正三角形, 正方形, 正五角形, あるいは最短の辺の長さが 1 である三角形である. これらを辺で接着した多角形で長さ 1 の線分を被覆することを考えると, 線分と共通部分を持つことができる多角形の個数は高々 4 つである (図 3.5). 従って, P の辺 e 上のスタンプの個数は高々 $4(|e|+1)$ である. 全ての辺に対して和をとれば $4(L+n)$ となる. 従って Array_Γ の要素数は $4(L+n)$ 以下である.

また, P の辺を分割しないスタンプ (つまり (3) で描画されるスタンプ) の個数は高々 $O(L+n)$ であるので, STAMP の要素数は全体で $O(L+n)$.

また, *Algorithm 1* の計算時間はスタンプが描画される回数に対して線形であるため, *Algorithm 1* の計算時間は $O(L+n)$ となる. \square

3.4 貼り合わせ確認

この節では与えられた貼り合わせ方 γ に対して, $\phi_\gamma(\partial P)$ が全域木かどうかを判定する方法について述べる.

貼り付け方は Q の表面上の座標を格納した配列 Array_Γ として与えられている. また切断木の葉に対応する P の周上の点へのポイントの配列 $\text{Array}_{\text{Leave}}$ が得られている. 全域性についてはすでにスタンプングのステップで確認されている. つまり, 確認すべきことは $\phi_\gamma(P_\gamma)$ が木構造をなしているかどうかである.

まず, Array_Γ を, 先頭と最後尾を繋いだサイクリックな双方向リスト List_Γ にコピーする. これに従って $\text{Array}_{\text{Leave}}$ も対応する List_Γ の要素を指す様に書き換

Algorithm 1: スタンピング

Input : 多角形 P を表す $Array_P$ とその初期位置 $(p = (0, 0), p' = (p'_x, p'_y))$,
多面体 Q を表す $List_Q$ とその初期位置 (q, e, f)

Output: Q の表面上の座標の配列 $Array_\Gamma$,
 $Array_\Gamma$ の要素へのポインタの配列 $Array_{Leave}$

Q の表面上の座標の配列 $Array_\Gamma$ を作成し, 空に初期化
 $Array_\Gamma$ へのポインタの配列 $Array_{Leave}$ を作成し, 空に初期化
 Q の各頂点に対応する配列 $Array_Q$ を作成し, 0 に初期化
 Q の面へのポインタ $base$ を作成し, f に初期化
多角形 P_{base} を作成し, (v, e, f) に従って $base$ を \mathbb{R}^2 に描画したものに初期化
 $i = 0$;

while $i < n$ **do**

- if** P の辺 (p_i, p_{i+1}) が P_{base} の周と交点 c を持つ **then**
 - c に対応する Q の表面上の点を s とする
 - $Coord(s)$ を $Array_\Gamma$ の最後尾に追加
 - if** s が Q の辺上の点 **then**
 - $e' := s$ を含む Q の辺
 - else**
 - $e' := base$ において s に時計回りに入射する Q の辺
 - s の $Array_Q$ の値を 1 にする
 - if** $\alpha(c) = \sigma(s)$ **then**
 - $Array_{Leave}$ に $Array_\Gamma$ の最後尾へのポインタを追加
 - end**
- end**
- $base := base$ と e' を共有する Q の面
- $P_{base} := e'$ の配置に従って $base$ を描画した平面上の多角形

else

- if** p_{i+1} が P_{base} に含まれる **then**
 - $i++$;
- else**
 - $q' := p_i$ に対応する Q の頂点
 - $e' := base$ において q' に時計回りに入射する Q の辺
 - $base := base$ と e' を共有する面
 - $P_{base} := e'$ の配置に従って $base$ を描画した平面上の多角形

end

end

end

if $Array_Q$ に値が 0 の要素がある, または $Array_\Gamma$ の先頭と末尾の値が一致しない **then**

- return** *false*

end

return *true*

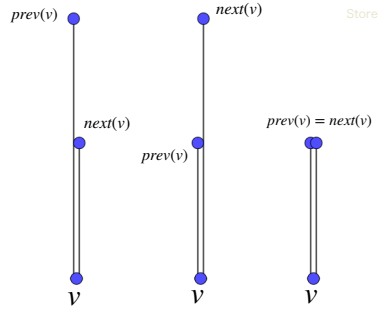


図 3.6: 葉が取りうる 3 つの状態

える. $List_\Gamma$ の各要素 v に対して, 前後の要素を $next(v), prev(v)$ とする.

$Array_{Leave}$ の先頭の要素を取り出し, これが指す $List_\Gamma$ の要素を v とする. v が切断木の葉である時, (1) $next(v)$ が線分 $(v, next(v))$ 上にある, (2) $prev(v)$ が線分 $(v, prev(v))$ 上にある, (3) $next(v) = prev(v)$ のいずれかが成り立つ (図 3.6). この要素を $List_\Gamma$ から削除する. これは切断木において葉 v を端点にもつ辺を削除することに対応する. さらに $next(v), prev(v)$ のうち v に近い元について (1),(2),(3) が成り立つかを確認し, 成り立つ場合それを $List_\Gamma$ から削除する. これを (1),(2),(3) が成り立たない要素が得られるまで続けることで, 切断木上で v から最も近い次数 2 以上の頂点までのパスを削除することができる. これを $List_\Gamma$ が空になるまで繰り返した結果, $List_\Gamma$ が空になれば $\phi_\gamma(P_\gamma)$ が木構造であり, そうでなければ木構造でないことがわかる. これによって $\phi_\gamma(\partial P)$ が木構造かどうかを確認することができる. 詳細は Algorithm 3 に示す.

ここで, 次の定理が成り立つ.

定理 28. 張り合わせ確認のアルゴリズムは $O(L + n)$ 時間で実行できる

Proof. 一つの要素に対して (1),(2),(3) が成り立つかどうかの確認は定数時間で行うことができ, 一度の確認で $List_\Gamma$ の要素は一つ削除される. 従って全体の計算時間は与えられた $Array_\Gamma$ の要素数に対して線形である. 従って $O(L + n)$. \square

3.5 Q が等面四面体の場合の対応点

本節では, Q が等面四面体である場合の対応点の探索について考える.

等面四面体の頂点は全ての曲率が π である. 従って, 葉によって切断された頂点の対応点は P の辺上に現れる. よって, すべての頂点が葉によって展開されている場合には対応点の候補を P の頂点から選ぶことができない. そこで多面体の頂点を, その上での切断木の状態に応じて分類し, それぞれの場合における対応点について考える.

Algorithm 2: 張り合わせ確認

Input : Q の表面上の座標の配列 $Array_{\Gamma}$,
 $Array_{\Gamma}$ の要素へのポインタの配列 $Array_{Leave}$

Output: 貼り付け写像が正しいかどうか
サイクリックな二重連結リスト $List_{\Gamma}$ を作成し, $Array_{\Gamma}$ をコピーする. これに従って $Array_{Leave}$ を書き換える

```
while  $Array_{Leave} \neq \emptyset$  do  
   $Array_{Leave}$  の最後尾から要素  $v$  を取る.  
  while  $(next(v) == (prev(v)))(Case1)$   
  または  $next(v)$  が線分  $(v, prev(v))$  上にある ( $Case2$ )  
  または  $prev(v)$  が線分  $(v, next(v))$  上にある ( $Case3$ ) do  
    if  $Case1$  then  
      |  $List_{\Gamma}$  から  $v$  を取り除く  
      |  $v := next(v)$   
    end  
    if  $Case2$  then  
      |  $List_{\Gamma}$  から  $v$  を取り除く  
      |  $v := next(v)$   
    end  
    if  $Case3$  then  
      |  $List_{\Gamma}$  から  $v$  を取り除く  
      |  $v := prev(v)$   
    end  
  end  
  if  $List_{\Gamma} = \emptyset$  then  
    | return true  
  else  
    | return fault  
  end  
end
```

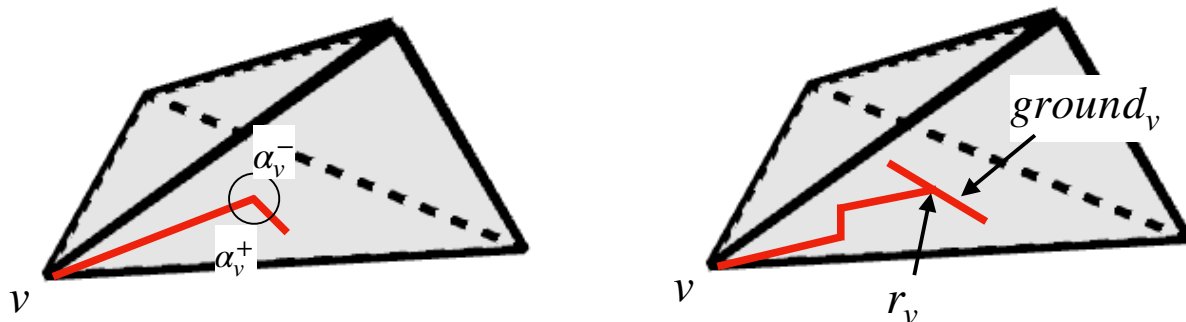


図 3.7: $e_v, r_v, \alpha_v^+, \alpha_v^-, g_v$ の定義

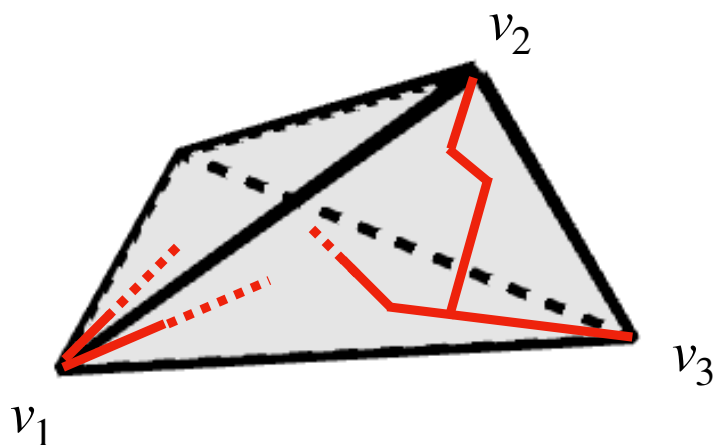


図 3.8: v_1 :分割頂点, v_2 :等分頂点, v_3 : v_2 に従属する準等分頂点

定義 29. $(cp(v), e_v, r_v, \alpha_v^+, \alpha_v^-)$ (図 3.7)

等面四面体 Q と切断木 CT に対して, Q の頂点 $v \in V(Q)$ の CT における次数が 1 であるとする.

- ・ v は P の周上に対応点を一つ持つので, これを $cp(v)$ で表す.
- ・ v を端点にもつ CT の辺を e_v とし, e_v の端点で v でない方を b_v した時, b_v を中心に e_v を反時計回りに回転させて隣の辺に重なるまでの角度を α_v^+ , 時計回りに回転させて隣の辺に重なるまでの角度を α_v^- で表す.
- ・ CT 上において, v に最も近い次数 3 以上の頂点を r_v とし, v から r_v までのパスを取り除いたグラフを CT_v とした時, r_v が CT_v の辺上の点であるならば, v はグラウンドを持つと呼び, その CT_v の辺を g_v で表す.

定義 30. (分割頂点, 等分頂点, 準等分頂点) (図 3.8)

等面四面体 Q が切断木 CT で展開されているものとし, $v \in V(Q)$ とする.

- ・ CT の v における次数が 2 以上である時, v を分割頂点と呼ぶ,
- ・ CT の v における次数が 1 であり, $\alpha_v^- \neq \pi$ かつ $\alpha_v^+ \neq \pi$ が成り立つ時, v を等分

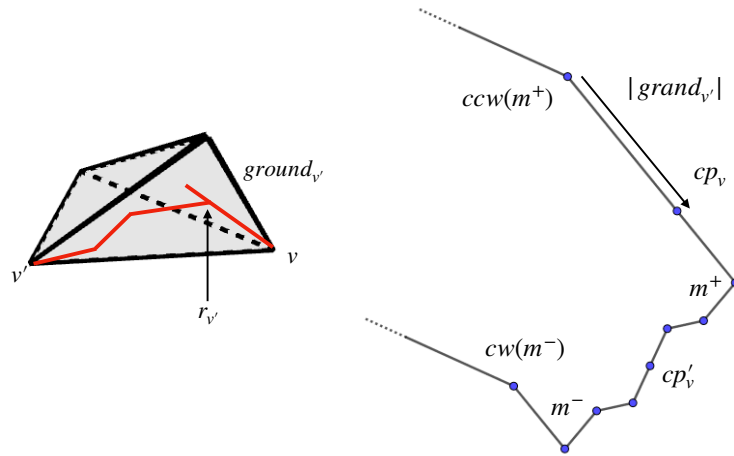


図 3.9: 準等分頂点の対応点

頂点と呼ぶ.

・ CT の v における次数が 1 かつ, グラウンドを持つ等分頂点 $v' (\neq v)$ が存在し, $g(v')$ が v を端点にもつ時, v を v' に従属する準等分頂点と呼ぶ.

補題 31. 等面四面体 Q が切断木 CT で展開されたとし, $v \in V(Q)$ とする.

つぎが成り立つ.

- ・ v が分割頂点であるならば, その対応点は全て $V(P)$ に含まれる.
- ・ v が等分点であるならば, その対応点 cp_v はある $e \in E(P)$ の中点に一致する.
- ・ v が v' に従属する準等分点であるとし, P の周上で $cp_{v'}$ を中心とする回転対称部分の両端点 m^-, m^+ を, P の周上に反時計回りに $m^-, cp_{v'}, m^+$ の順で現れる様に取り (図 3.9). この時, $(cw(m^-), m^-)$ と $(m^+, ccw(m^+))$ のどちらか長い方の辺上に cp_v があり, その辺の 2 つの端点のうち $cp_{v'}$ から遠い方 ($cw(m^-)$ または $ccw(m^+)$) と cp_v の距離は $|grand_{v'}|$ である.

Proof. v が分割頂点である場合について考える, v における CT の次数は 2 以上なので, v は 2 つ以上の対応点を持つ. また, $\sigma(v) = \pi$ より, 対応点の内角の和は π であるので, 対応点の内角は全て π 未満である. 従って, 全ての対応点の内角は π でない.

つぎに, v が等分点である場合について考える. $\alpha_v^- \neq \pi$ かつ $\alpha_v^+ \neq \pi$ より, $cp(v)$ はこれらの内角を持つ P の辺の中点に現れる.

最後に, v が v' に従属する準等分点である場合について考える (図 3.9). CT の v' から $r'(v')$ までのパスは, P の周上で $cp_{v'}$ を中心とする線対称部分を形成する. この両端点を m^-, m^+ とする. $g_{v'}$ 上の点は, P の周上で $(cw(m^-), m^-)$ または $(m^+, ccw(m^+))$ 上の点对応する. よって, $|cw(m^-), m^-| + |m^+, ccw(m^+)| = 2|g_{v'}|$ が成り立つ. 従って, cp_v は $(cw(m^-), m^-)$ と $(m^+, ccw(m^+))$ のどちらか長い方に現れ, $cp_{v'}$ から遠い方の頂点からの距離は $|g_{v'}|$ □

ここで次の補題が成り立つ.

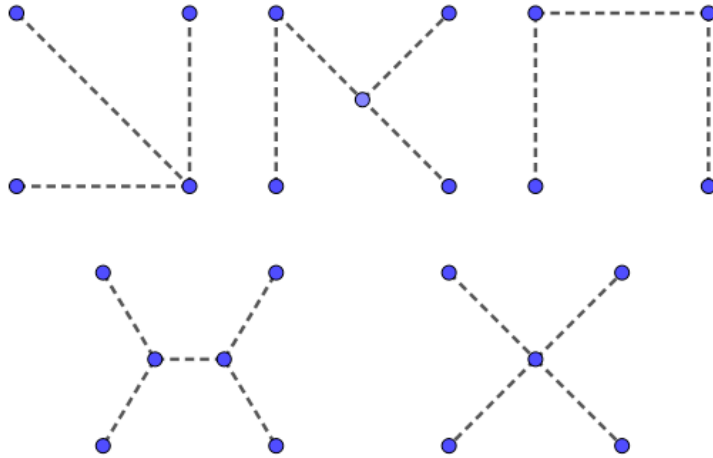


図 3.10: 等面四面体の切断木の分類, 破線はパスの略記

補題 32. 多角形 P が等面四面体 Q の展開図であるとする. この時, P を与える切断木 CT で次の条件のいずれかを満たすものが存在する.

- ・ Q は 1 つ以上の分割頂点をもつ.
- ・ Q は 2 つ以上の等分頂点をもつ.
- ・ Q は 1 つの等分頂点とそれに従属する準等分頂点を持つ.

Proof. Akiyama 等の結果 [10] によると, 等面四面体の切断木 CT は (図 3.10) の 5 つに分類できる.

Q が次数 1 でない頂点を持つ場合 (図 3.10 上段), それらは分割頂点である, そこで, 全ての頂点において次数が 1 である場合 (図 3.10 下段) について考える.

Q の頂点 v が, v から r_v までの CT 上のパス上に次数 2 の頂点 b_v を持つ時, α_v^- か α_v^+ のどちらかが π であるとする, b_v が次数 2 の Q の頂点となり, 全ての頂点の次数が 1 であることに矛盾する. 従って $\alpha_v^- \neq \pi$ かつ $\alpha_v^+ \neq \pi$ だが成り立ち, v は等分頂点である. つまり, (*) 全ての頂点において v と r_v は辺を共有する, という仮定のもと主張が成り立つことを確認すれば十分である. 以降は (*) を仮定する.

全ての頂点において次数が 1 である場合 (図 3.10 下段) は次の 2 つの場合に分類できる.

(1) CT が次数 4 の頂点 r を 1 つ持つ場合

この場合, $r = r_{v_1} = r_{v_2} = r_{v_3} = r_{v_4}$ が成り立つ. r の周りの 4 つの角の内, 少なくとも 3 つの角度は π でない. この 3 つの角の間の 2 つの辺が入射する Q の 2 つの頂点はそれぞれ等分頂点である. 従ってこの場合には Q は 2 つの等分頂点を持つ.

(2) CT が次数 3 の頂点 r_1, r_2 を持つ場合

この場合, $r_1 = r_{v_1} = r_{v_2}, r_2 = r_{v_3} = r_{v_4}$ が成り立つとしても一般性を失わない. ここで e_{v_i}, e_{v_j} のなす角を $\angle(e_{v_i}, e_{v_j})$ と表すとする. r_1 の周りの 3 つの角の内, 少

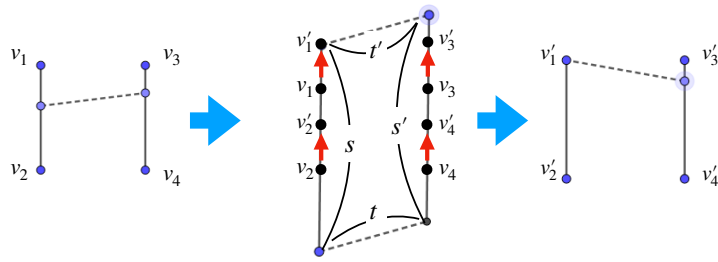


図 3.11: $\angle(e_{v_1}, e_{v_2}) = \pi$ かつ $\angle(e_{v_3}, e_{v_4}) = \pi$ の場合

なくとも2つの角度は π でないことから、 $\angle(e_{v_1}, e_{v_2}) \neq \pi$ ならば、 v_1 か v_2 の少なくとも一方は等分頂点である。同様に $\angle(e_{v_3}, e_{v_4}) \neq \pi$ ならば、 v_3 か v_4 の少なくとも一方は等分頂点である。したがって、 $\angle(e_{v_1}, e_{v_2}) \neq \pi$ かつ $\angle(e_{v_3}, e_{v_4}) \neq \pi$ ならば、 Q は2つの等分頂点を持つ。それ以外の場合を次の2つに分けて考察する。

(2-1) $\angle(e_{v_1}, e_{v_2}) = \pi$ かつ $\angle(e_{v_3}, e_{v_4}) = \pi$ の場合

この場合、この切断木によって得られる展開図 P は平行で長さの等しい2つの辺 s, s' を持ち、 P の周りの残りの2つの部分 t, t' は互いに並進対称である(図3.11)。この時、 P は s を二等分して、 t, t' を貼り合わせるようにしても Q を折ることができる。この折り方を与える切断木を CT' とすると、 CT' は分割頂点を持つ。

(2-2) $\angle(e_{v_1}, e_{v_2})$ と $\angle(e_{v_3}, e_{v_4})$ のどちらか一方が π 、他方が π でない場合

$\angle(e_{v_1}, e_{v_2}) \neq \pi$ と仮定しても一般性を失わない、さらに、 v_1, v_2 の少なくとも一方は等分頂点であることから、 v_1 が等分頂点であるとしても一般性を失わない。

この時、 r_1 における3つの角のうち、 e_{v_1} 以外の2辺からなる角度が π でなければ r_2 も等分頂点である。これが π であるときには、 v_1 から r_1 までを CT から取り除いたグラフ CT_{v_1} において、 r_1 は CT_{v_1} の辺上の点であり、その辺は g_{v_1} に一致する。 g_{v_1} は v_2 を端点にもつので、 v_2 は v_1 に従属する準等分頂点である。□

以上の考察より、対応点の探索をAlgorithm 3によって行うことができる。2つの対応点が $V(P)$ に含まれる場合、その組合せは n^2 通りであり、この列挙は $O(n^2)$ 時間で実行可能である。2つの対応点が $E(P)$ の中点に含まれる場合も同様である。それ以外の場合、一つの対応点には n 通りの選び方があり、それぞれに対して回転対称部分の両端点の探索は $O(n)$ 時間で実行可能であることより $O(n^2)$ 時間で計算可能である。

3.6 対応点の位置の列挙

これまでに、多角形 P の配置は周上の点 p, p' を選んで、その座標($p = (0, 0), p' = (p'_x, p'_y)$)は $p_x'^2 + p_y'^2 = |p, p'|^2$ を満たすことを確認した。この節では p' の座標(p'_x, p'_y)を定める方法について考える。

Algorithm 3: 等面四面体の対応点探索

Input : $Array_P$ で表される多角形 $P, List_Q$ で表される多面体 Q

Output: P の周上の二点の組

for each $v, v' \in V(P)$ **do**

 | $Output(v, v')$

end

for each $e, e' \in E(P)$ **do**

 | m, m' を e, e' の中点とする.

 | $Output(m, m')$

end

for each $e \in E(P)$ **do**

 | m を e の中点とする.

 | m 中心の線対称部分の両端点を m^-, m^+ とする.

 | **if** $\alpha(m^-) + \alpha(m^+) = \pi$ **then**

 | **if** $|m^+, ccw(m^+)| > |m^-, cw(m^-)|$ **then**

 | m' を $ccw(m^+)$ から時計回りに $\frac{|m^-, cw(m^+)| + |m^+, ccw(m^+)|}{2}$ だけ進んだ点とする.

 | **else**

 | m' を $cw(m^-)$ から反時計回りに $\frac{|m^-, cw(m^+)| + |m^+, ccw(m^+)|}{2}$ だけ進んだ点とする.

 | **end**

 | $Output(m, m')$

 | **end**

end

p から p' までスタンピングを行ったと考える. p' が対応点であることから, p' には平面上でいずれかのスタンプの頂点に対応する. そこで次の概念を定める.

定義 33. (格子点集合)

多面体 Q の初期位置を (v, e, f) とする. この時, 何らかの初期配置で置かれた多角形 P の上でのスタンピングを行った時の *STAMP* の頂点になりうる平面上の点の集合を Q の $((v, e, f)$ で生成された) 格子点集合と呼び, $Lattice(Q_{v,e,f})$ で表す.

つまり, p' の位置の候補は $C \cap Lattice(Q_{v,e,f})$ に含まれる. これを全て列挙することで, それらを p' の候補とすることができる.

定義 34. (基本ベクトル)

初期位置 (v, e, f) の多面体 Q に対して, 基本ベクトル $Base(Q_{v,e,f})$ を次で定める.

- (1) $Q \in \text{IntBox}$ の時, $Base(Q_{v,e,f}) := \{\vec{b}_0 = (1.0), \vec{b}_1 = (0, 1)\}$
- (2) $Q \in \text{HiDelta}$ の時, $Base(Q_{v,e,f}) := \{\vec{b}_0 = (1.0), \vec{b}_1 = (0, \frac{\sqrt{3}}{2})\}$
- (3) $Q \in \text{Tetramono}$ の時, $Base(Q_{v,e,f}) := \{\vec{b}_0, \vec{b}_1; \text{原点から } f \text{ の } v \text{ 以外の } 2 \text{ つの頂点 } v_1, v_2 \text{ それぞれへのベクトル}\}$
- (4) $Q \in \text{Dodeca}$ の時, $Base(Q_{v,e,f}) := \{\vec{b}_i = (\cos \frac{2\pi i}{5}, \sin \frac{2\pi i}{5}), (i = 0, 1, 2, 3, 4)\}$

ここで次の定理が成り立つ.

定理 35. (基本ベクトルによる格子点集合の生成)

$Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Dodeca} \cup \text{Tetramono}$ とし. $\vec{b}_i \in Base(Q_{v,e,f})$ とする.

この時, 任意の $Lattice(Q_{v,e,f})$ の元 l に対して, $l = \sum_i x_i \vec{b}_i$ を満たす整数 $-4(L+n) < x_i < 4(L+n)$ が存在する.

Proof. Q を初期位置 (v, e, f) に置いた時, v から f の各頂点へのベクトルは $Base(Q_{v,e,f})$ の元を高々一つ使って表せる. f の辺 e' を任意に選んで, その上にスタンプ s を描画した場合を考える. e' の一方の端点を v' とすると, v' から s の各頂点へのベクトルも $Base$ の元を高々一つ使って表せる (図 3.12). 従って p, p' を結ぶベクトルは, $Base(Q_{v,e,f})$ の元の整数線型結合で表すことができる. この係数は P の辺を分割しているスタンプの個数を超えないため, $\pm 4(L+n)$ の範囲である. \square

以上の考察より, P の周上の 2 点 p, p' が対応点の候補として与えられた時, それらの平面上の座標の候補の列挙は Algorithm 4 によって行うことができる. ここで, p' が C に含まれることより, $Base(Q_{v,e,f})$ の係数のうち一つ以外が固定されると, 残り一つの係数は 2 通りの値に限定され, これは定数時間で計算できる. 従って, Algorithm 4 は, $Q \in \text{Dodeca}$ の場合に $O((L+n)^3)$ 時間, その他の場合は $O((L+n))$ 時間で実行できる.

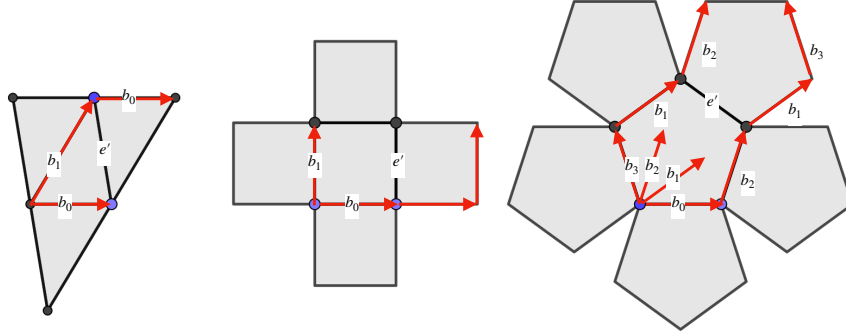


図 3.12: 対応点間のベクトル

Algorithm 4: 対応点の座標の列挙

Input : $Array_P$ で表された P , P の周上の二点 (p, p') ,
 $List_Q$ で表された多面体 Q , Q の初期位置 (v, e, f)

Output: p' の座標 $p' = (p'_x, p'_y)$

$C :=$ 原点中心, 半径 $|p, p'|$ の円

if $Q \in \text{Dodeca}$ **then**

 | $i := 3$

else

 | $i := 1$

end

for $-4(L+n) \leq x_0, \dots, x_{i-1} \leq 4(L+n)$ **do**

 | $\vec{v} := \sum_{j=0}^{i-1} x_j \vec{b}_j$, $m := \vec{v} + t\vec{b}_i$

 | $d_0, d_1 := C$ と m の 2 つの交点それぞれへの \vec{v} の終点からの距離

 | **for** $k = \{0, 1\}$ **do**

 | **if** $d_k \in \mathbb{Z}$ **then**

 | Output $(p = (0, 0), p' = (\sum_{j=0}^{i-1} x_j \vec{b}_j) + d_k \vec{b}_i)$

 | **end**

 | **end**

end

3.7 まとめ

本研究では $Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Dodeca} \cup \text{Tetramono}$ の場合の折り判定問題を解く、多項式時間アルゴリズムを与えた。

まず、 Q が等面四面体の場合は Algorithm 3 によって、それ以外の場合は頂点2つの組み合わせによって、対応点の組の候補を列挙する。これは $O(n^2)$ 時間で実行できる。次に、それぞれの対応点の組に対して Algorithm 4 でその座標を定め、 P の位置を決定する。これは $Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Tetramono}$ の場合は $O(n+L)$ 時間、 $Q \in \text{Dodeca}$ の場合 $O((n+L)^3)$ 時間で実行可能である。さらに、 Q のそれぞれの初期配置に対して Algorithm 1 によってスタンプングを行い一つの張り合わせ方を得る。最後に、得られた張り合わせ方が全域木を正しく与えるかどうかを Algorithm 2 によって確かめる。従って、アルゴリズム全体に要する計算時間は $Q \in \text{IntBox} \cup \text{HiDelta} \cup \text{Tetramono}$ の場合は $O(n^2(n+L)^2)$ 、 $Q \in \text{Dodeca}$ の場合は $O(n^2(n+L)^5)$ である。

最後に、先行研究と本研究を比較する。 $Q \in \text{IntBox}$ の場合について、先行研究 [6] は $O(D^7 n^2 (D^5 + \log n))$ 時間の折り判定アルゴリズムを示している。ここで D は多角形の頂点間の距離の最大値を表す。ここで $2D \leq L$ が成り立つので、本研究のアルゴリズムは既存のアルゴリズムを真に高速化していると言える。 $Q \in \text{HiDelta}$ の場合について、先行研究 [7] は $O((f+n+L)n^3 mL \log n)$ 時間の折り判定アルゴリズムを示している。これに対しても本研究で提案したアルゴリズムは真に高速なアルゴリズムになっている。また [7] のアルゴリズムは、多面体の対象範囲が HiDelta の中でも、曲率 π の頂点を持たない場合に限られている。本研究ではこの例外条件を取り除き、全ての HiDelta に適用できる形に一般化した。

$Q \in \text{Tetramono}$ の場合については、先行研究 [4],[3] は Tetramono の展開図を幾何学的に特徴付けを与えている。本研究では、この結果に折り判定アルゴリズムのフレームワークを組み合わせることで、他の多面体の場合と同様に折り判定が行えることを示した。

第4章 Future Work

本論文で述べたアルゴリズムをより広い多面体のクラスに拡張するためには, 3.6 節において *Lattice* の元を列挙する部分に新たな方法論が求められる. Q が *IntBox*, *HiDelta*, *Tetramono* または *Dodeca* に含まれるとき, 面の対称性によって *Lattice* を生成する定数個のベクトルの集合 *Base* が存在する. 一般にはこのようなベクトルの集合は存在しない. 例えば, 菱形十二面体などの π の有理数倍でない曲率を持つ頂点がある多面体の場合には, 新たな手法が求められる

第5章 謝辞

本研究を行うにあたり，研究テーマの設定から論文作成までご指導をいただいた，主指導教員の上原隆平教授に深く御礼申し上げます。また，共同研究を通して様々な助言をくださった北海道大学の堀山貴史教授，審査会を通してご助言をくださった審査員の金子峰雄教授，緒方和博教授，廣川直准教授にも御礼申し上げます。また折に触れて様々なご指摘をいただいた上原研究室の皆様，特に発表や論文作成の英語の添削をしてくださった谷口智子研究補助員に御礼申し上げます。

最後に，日頃から絶え間ない支援と応援を下さった家族とパートナーに改めて深い感謝を申し上げます。

参考文献

- [1] ユークリッド原論, (著) ユークリッド, (訳, 解説) 中村幸四郎, 寺坂英孝, 伊東俊太郎, 池田美恵. 共立出版, <TW00005298>. (1971)
- [2] A. Dürer. Underweysung der messung, mit den zirckel un richtscheyt, in Linien ebnen unnd gantzen corporen (1525)
- [3] Akiyama, J.: Tile-Makers and Semi-Tile-Makers. The Mathematical Association of America, Monthly 114, pp. 602-609, August-September 2007
- [4] Akiyama, J., Nara, C.: Developments of polyhedra using oblique coordinates. J. Indonesia. Math. Soc. 13(1), 99-114 (2007)
- [5] Horiyama, T., Mizunashi, K.: Folding orthogonal polygons into rectangular boxes. In: Proceedings of the 19th Japan-Korea Joint Workshop on Algorithms and Computation (WAAC 2016) (2016)
- [6] Mizunashi K., Horiyama T., Uehara R. (2019) Efficient Algorithm for Box Folding. In: Das G., Mandal P., Mukhopadhyaya K., Nakano S. (eds) WALCOM: Algorithms and Computation. WALCOM 2019. Lecture Notes in Computer Science, Vol 11355. pp,277-288, Springer(2019)
- [7] 門口 あきら, 凸な高次デルタ多面体を対象にした折り判定問題, 修士論文 (2020)
- [8] Kamata, T., Kadoguchi, A., Horiyama, T., Uehara, R.: Efficient Folding Algorithms for Regular Polyhedra. In: 32nd Canadian Conference on Computational Geometry (CCCG 2020), pp. 131-137 (2020).
- [9] E. D. Demaine and J. O'Rourke. Geometric Folding Algorithms: Linkages, Origami, Polyhedra. Cambridge University Press, 2007.
- [10] Jin Akiyama, Kyoko Matsunaga. Every Conway tile can be folded into an isotetrahedron. unpublished.
- [11] 秋山仁, 離散幾何学フロンティア, 近代科学社 (2020)

- [12] Mitani, J., Uehara, R.: Polygons Folding to Plural Incongruent Orthogonal Boxes. In: 20th Canadian Conference on Computational Geometry (CCCG 2008), pp. 13-15 (2008).