

極大凸多角形集合による平面被覆問題の解の全列挙と最適解の抽出手法

○瀧澤 重志*¹

キーワード：Space Syntax，可視グラフ，極大凸多角形，極大クリーク列挙，極小横断列挙

1. はじめに

Bill Hillier らが考案した Space Syntax¹⁾ (以下 SS) は、建築や都市の空間構成を定量的に分析する手法として現在世界中で普及が進んでいる。SS の分析手法の中でよく用いられているのは Axial map である。これは空間の幾何的な特徴で決定される視線のネットワークとして空間を分節化し、Integration value や Depth などのネットワーク分析の指標によって空間のつながりを特徴づけようとするものである。この手法が提案された当初は、Axial map は手作業で作成されていたが、計算機への実装が考えられるようになると、その理論的背景が弱いことがたびたび指摘されてきた。Hillier の定義によれば、Axial map は空間全体を監視できる最小個数かつ最長の視線の集合とされていた。しかしこの言葉通りにアルゴリズムを実装しようとしても、人が手で描いたような図を作ることが難しいことがわかってきた。

これらの問題に対応するために、1990年代後半から、何人かの著名な研究者によって Axial map を自動生成するアルゴリズムが提案されてきた。例えば Peponis らは、平面を凸領域に分割し、それらを全て横断し、かつ連結制約とトポロジカル制約を満たす最小の視線集合を求めるヒューリスティクスを開発した²⁾。また、Turner らは、平面内のオブジェクトの各頂点を結んだすべての視線の集合から、極大なものだけを選ぶ Subset elimination と呼ばれる手法を用いて視線を間引き、さらにいくつかのヒューリスティクスを加えて Axial map を作成する方法を提案した³⁾。しかしいずれの方法も近似解法にとどまっている。なぜならばこれらの問題は、空間全体を見渡す最小の視線や領域の組み合わせを求めることが本質であるが、これらは集合被覆問題と呼ばれる NP 困難な問題に属するからである。加えて、もともとの凸分割を求める問題も NP 困難である。さらに、空間の被覆・凸分割のパターンも一通りではなく、複数存在し得る。これらに加えて、視線の接続関係やトポロジカルな関係の制約が問題を一層困難にしている。関連して、現代の空間は分節が曖昧になってきているが、そうした空間に対して Axial map のような線で置き換えるタイプの分析手法は向いているとはいえない。

一方計算機科学の分野では、こうした計算困難な問題に対して、計算機の性能向上や新たなアルゴリズムの開

発などで、ある程度の規模の問題であれば、実時間で厳密解を求められるほど技術革新が進んでいる。その中で本研究では列挙アルゴリズムに着目する。代表的なアルゴリズムに、バックトラック、逆探索、ZDD⁴⁾を利用したものなどの手法がある。本研究で扱う列挙問題は、極大クリーク列挙とハイパーグラフの極小横断列挙と呼ばれるもので、いずれも実用規模のデータに対する高速なアルゴリズムが提案されている。

上記の背景から本研究では、Axial map の抱える問題を解決し、従来の線的なつながりではなく、まとまりのある領域のつながりとして空間全体を分節化するために、極大凸多角形による平面被覆問題を定義し、最新の高速列挙アルゴリズムを用いて全解列挙し、最適解を抽出する方法を提案する。

2. 提案手法

本研究での平面被覆問題は、平面上に配置されたポリゴンの隙間の空間を、くまなく極大な凸多角形によって被覆するための最小個数の凸多角形の組み合わせを求める問題とする。なおポリゴンの線分はすべて直線とする。

2.1 空間の分割

2.2 で示すように、入力されたポリゴンデータの頂点を結んで可視グラフを作り、3.3 でそれらのクリーク、すなわち凸な多角形となる可視領域を列挙する。図1は、正方形の平面内に4つの正方形がある空間における可視領域のタイプを示している。Aは存在するオブジェクトの端点を結んでできる可視領域で、有限個数ある。一方、B、Cは、その領域の端点の一部もしくはすべてが、オブジェクトの端点を通らないタイプの可視領域であり、これらは無限個存在する。B、Cのタイプの凸多角形は、無

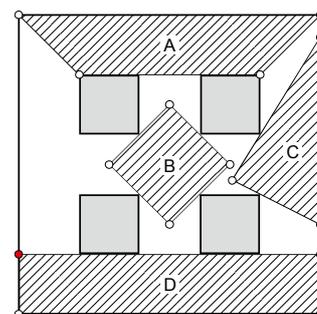


図1 可視領域のタイプ

限個考えられることと、まとまった空間として認知されにくいと思われるので、本研究では考慮しない。一方 A のタイプの可視領域だけではバリエーションが少ないことと、空間認知の観点から、次の考え方で拡張する。Peponis は、壁面を延長して、それが他の壁面と交わる点を結んで空間を分割する手法を提案し、**s-partition** と名付けた⁵⁾。例えば、図 1 における領域 D の赤丸の頂点は **s-partition** によって分割されて新たに加えられた頂点である。s-partition は壁面線の境界で空間を分割するが、この場所は空間の可視性が大きく変化するので、空間の認知に大きな影響を与える場所と考えられる。そこで本研究では s-partition によって壁面線を分割し、分割された線分の頂点と、もともと入力されたポリゴンの頂点からなる集合を V として、可視グラフを作成する。

2.2 可視グラフの作成

V の全ての頂点ペア $u, v \in V, u \neq v$ について、それらを端点とする線分を引き、それが他の壁面線と交わらない場合、その線分を残す。そして、それらの線分集合を E とする可視グラフを $G = (V, E)$ とする。

2.3 極大クリーク（極大凸多角形）の列挙

可視グラフ G の頂点が 3 点以上のクリークはお互いに可視な部分グラフ、すなわち凸多角形となる。例えば図 2 のように、矩形の障害物が置かれている平面において、頂点 $\{a, b, c, d, e\}$ からなる可視グラフが描かれているとする。この可視グラフには、頂点 $\{a, b, c\}, \{a, b, d\}, \{b, c, d\}, \{c, d, e\}, \{a, b, c, d\}$ からなる五つの（頂点が三つ以上の）クリークが含まれている。これらのうち頂点 $\{a, b, c\}, \{a, b, d\}, \{b, c, d\}$ のクリークは、 $\{a, b, c, d\}$ のクリークの部分グラフとなっているので無視できる。そうして残った頂点 $\{a, b, c, d\}$ と $\{c, d, e\}$ からなるクリークを**極大クリーク**と呼ぶ。 G からこれら極大クリークのみを、既存の極大クリーク列挙手法を使用して列挙する。本研究では ZDD をベースとした Coudert のアルゴリズム⁶⁾を使用する。列挙された極大クリークの集合を C 、それから構成される凸多角形の集合を $P(C)$ とする。

2.4 サンプルング点の設定

C の要素によって平面を被覆する。平面をくまなく被覆するためには、計算幾何学の**アレンジメント**の問題を解く必要がある。アレンジメントとは、互いに交差する超平面によって分割された空間構造のことである。本研究では得られた極大凸多角形の線分によって平面を分割することに相当する。しかし、アレンジメントの計算は計算量が線分の数の 2 乗のオーダーで増え、極端に小さい閉領域が発生しやすくなる。これらは建築的に意味のある大きさではなく、それらが増えれば、2.5 の列挙で計算爆発を起こしやすくなる。

そこで本研究では、空間内に細かくサンプルング点を設け、それらを被覆することで目的を達成する。計算の

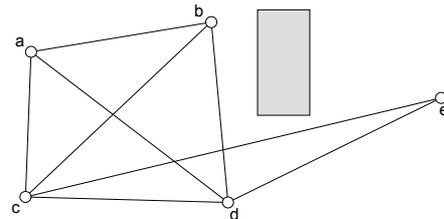


図 2 可視グラフとクリーク

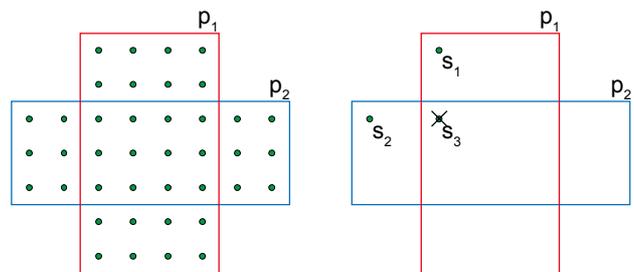


図 3 サンプルング点全体 (左) と判定に必要な点 (右)

複雑度は、極大凸領域の数に依らず、サンプルング点数のオーダーに固定される。ここでは間隔 l の格子グリッド上で点を発生させる。これらの点すべてを使う必要はなく、2.5 の計算で必要な点集合に間引き、それを S とおく。例えば図 3 左は二つの凸多角形 p_1, p_2 内に等間隔に点を発生させた例であるが、2.5 の計算には、 s_1, s_2 の 2 点があれば十分である。これは、各点からみて、その点が p_1, p_2 の各領域に含まれるか否かを計算し、含まれる領域のパターンが既に採用された他の点のパターンと同じであれば、その点は捨てるという単純なアルゴリズムで判定できる。さらに、この例の場合、上記のアルゴリズムで s_3 も抽出されるが、 s_3 は p_1 と p_2 の両者に含まれるので不要と判断する。なぜなら、 s_1, s_2 が含まれる領域はそれぞれ p_1, p_2 だけであるから、 p_1 と p_2 は必ず使われる。したがって、両者に含まれる s_3 は冗長な点となる。逆に s_3 のみを残すと、 p_1 と p_2 のいずれかを残せば s_3 が被覆できるので、 s_1, s_2 のいずれかが被覆されなくなる。

2.5 最小個数の凸多角形の抽出

図 4 のように、サンプルング点 $s_1 \sim s_{10} \in S$ と凸多角形 $p_1 \sim p_5 \in P(C)$ があるとすると、このとき、表 1 のように、各サンプルング点を含む凸多角形の部分集合の組み合わせを E' とすると、 $H = (P, E')$ を**ハイパーグラフ**、 E' を**ハイパーエッジ**と呼ぶ。このとき、すべての E' と交わる P の部分集合を H の**横断 (hitting set)**、横断の中で極小なものを**極小横断**と呼ぶ。本研究の問題の場合、極小横断とはサンプルング点を包含する凸多角形の極小な組み合わせのことである。この例の場合、極小横断は、 $\{p_2, p_5\}$ と $\{p_3, p_4, p_5\}$ の二組存在する。

この極小横断の中に求めたい最小個数の凸多面体の組み合わせがある。集合の極小横断を求める問題と最小

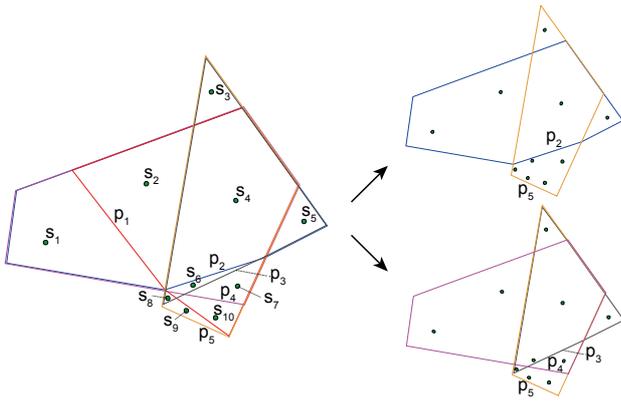


図4 サンプルング点を包含する凸多角形の極小横断

表1 図4において各点を包含する凸多角形

点	包含する凸多角形	点	包含する凸多角形
s_1	$\{p_2, p_4\}$	s_6	$\{p_1, p_3, p_4, p_5\}$
s_2	$\{p_1, p_2, p_4\}$	s_7	$\{p_1, p_4, p_5\}$
s_3	$\{p_3, p_5\}$	s_8	$\{p_3, p_5\}$
s_4	$\{p_1, p_2, p_3, p_4, p_5\}$	s_9	$\{p_5\}$
s_5	$\{p_2, p_3\}$	s_{10}	$\{p_1, p_5\}$

の被覆を求める問題は等価であり、NP 困難である。しかし近年、こうした問題に対しても、ある程度の規模までならば効率的に解けるアルゴリズムが開発されており、本研究ではそれらの一つであるハイパーグラフの極小横断の列挙アルゴリズムを利用する。現在のところ、最も効率的なアルゴリズムといわれているのは戸田のアルゴリズム⁷⁾であるので、それを実装して利用した。

このアルゴリズムを使うことで、 H を入力し、その極小横断の集合を ZDD の形で取得できる。本研究で求めたいのは最小個数（極小と意味が違うことに注意）の凸多角形の組み合わせで、かつそれらの中で面積の和が最大なものである。一般に極小横断は膨大な数が得られるので、それらを一つ一つ展開してチェックするのは困難である。そこで、ZDD のデータを圧縮したままライブラリ (Graphillion⁸⁾) の集合演算の機能を利用して効率的に解を求める。まず最小個数の組み合わせの集合を PermitSym 関数で取得し、それらの中から重みを各凸多角形の面積として、Choose_Best 関数で重み最大の組み合わせを抽出する。この他に、面積の大きい凸多角形から辞書式順序で組み合わせを抽出することも簡単にできる。

3. 検証

検証に用いた空間は、藤本壮介による情緒障害児短期治療施設のフロアプランを模したものである (図5)。個室群がオープンスペースでルーズに接続された構成となっている。図6が計算対象となる空間のみを抜き出した領域で、図7~11が2章で示したプロセスに対応し、図12が最終的に得られた被覆である。この例ではサンプリ

ング点の間隔 $l = 10\text{cm}$ として計算した。人間の目で見ても、違和感なく凸多角形の被覆ができていると思われる。

計算には一般的なノートパソコン (VAIO Z, CPU: Core i7-5557U, memory: 16GB, OS: Windows 8.1 Professional, compiler: Microsoft Visual C++2015) を用いて行った。得られた主なオブジェクトの個数は以下の通りである。可視グラフのノード数: 179, 同エッジ数: 1728, 得られた極大クリークの数: 219, うち壁無しクリークを排除したもの: 216, サンプルング点の数: 1072, 得られた極小横断の数: 1543606641113586432 (1543 京), 最小個数の極小横断の数: 凸多角形 35 個のものが 18480 通り。さらに計算にかかった時間等は以下の通りである。極大クリーク列挙: 0.05 秒, 極小横断列挙: 0.5 秒, 最小個数の極小横断集合の探索: 1 秒, 最大面積の極小横断の探索: 0.01 秒, 使用メモリ: 70MB。

4. まとめ

本研究では、SS の抱える問題点を解消するために、凸多角形による平面被覆問題を定義し、その厳密解法を提案し検証を行った。その結果、高速に膨大な組み合わせと厳密解を求められることを確認した。今後の課題として、大規模な空間に適用する為のアルゴリズムの改良・数値計画ソルバの併用や、ハイパーグラフを活用した空間の評価指標の構築などがあげられる。

謝辞

本研究は科学研究費補助金基盤研究 C(16K06652)の補助の下で行われました。

[参考文献]

- 1) B. Hillier and J. Hanson, The Social Logic of Space. Cambridge University Press, 1984
- 2) J. Peponis, et al., On the generation of linear representations of spatial configuration, Environment and Planning B: Planning and Design, 25, pp.559-576, 1998
- 3) A. Turner, A. Penn and B. Hillier, An algorithmic definition of the axial map, Environment and Planning B: Planning and Design, 32, pp.425-444, 2005
- 4) S. Minato, Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, in Proc. of the 30th International Design Automation Conference, pp.272-277, 1993
- 5) J. Peponis, et al., On the description of shape and spatial configuration inside buildings: convex partitions and their local properties, Environment and Planning B: Planning and Design, 24, pp.761-781, 1997
- 6) O. Coudert, Solving graph optimization problems with ZBDDs, Proceedings of European Design and Test Conference 1997, pp.224-228, Mar 1997
- 7) T. Toda, Hypergraph Transversal Computation with Binary Decision Diagrams, in Proc. of 12th International Symposium on Experimental Algorithms (SEA2013), LNCS 7933, Rome, Italy, pp.91-102, 2013
- 8) T. Inoue, et al., Graphillion: Software Library Designed for Very Large Sets of Labeled Graphs, International Journal on Software Tools for Technology Transfer, 18(1), pp.57-66, Feb. 2016

*1 大阪市立大学工学研究科 准教授 博士 (工学)

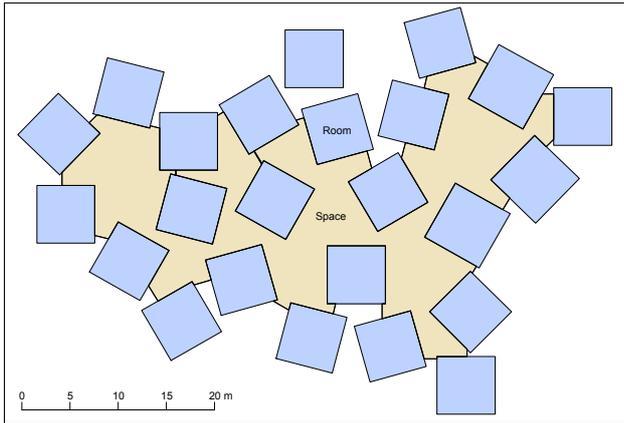


図5 対象としたフロアプランの略図

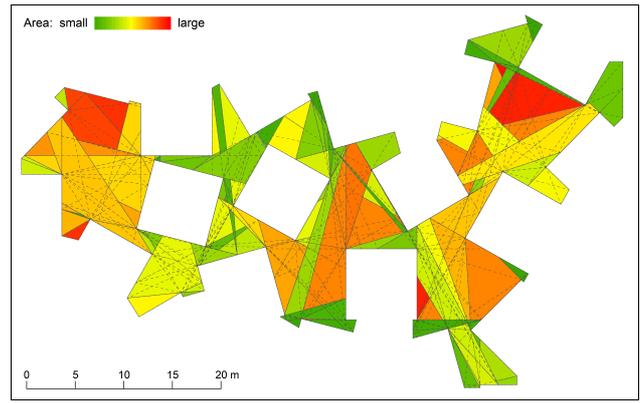


図9 列挙された極大凸多角形の集合

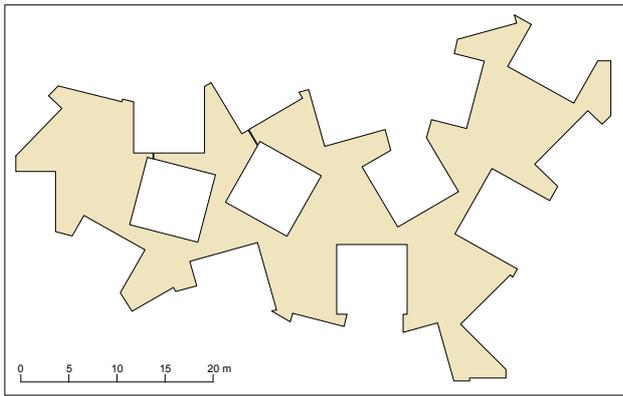


図6 空間のみを抽出した領域

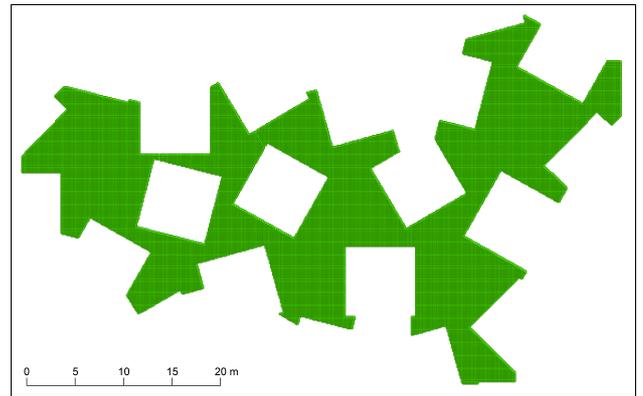


図10 すべてのサンプリング点

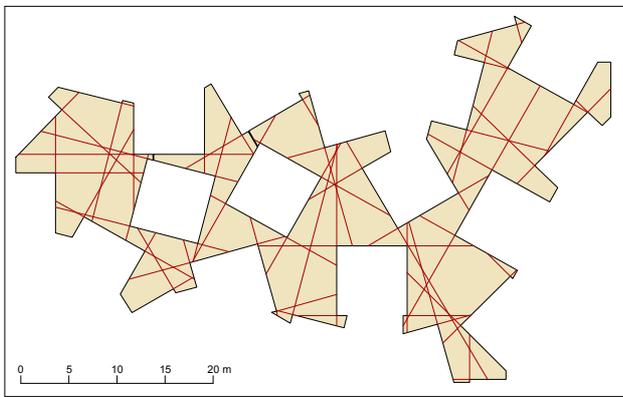


図7 s-partition

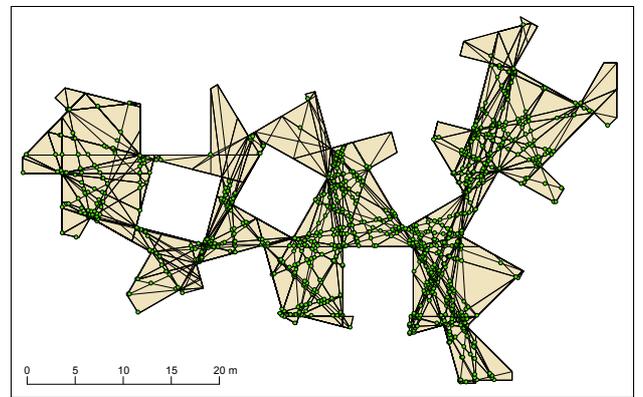


図11 被覆の判定に必要なサンプリング点

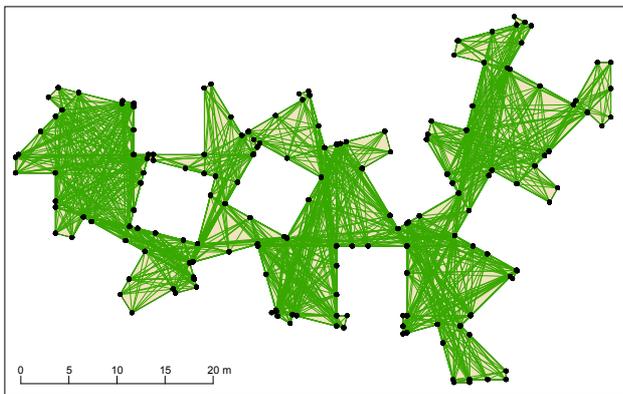


図8 可視グラフ

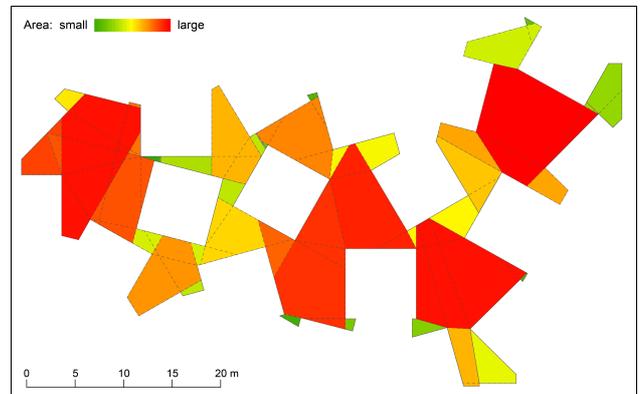


図12 最終的に得られた被覆