[IROS 2016]

The Constriction Decomposition Method for Coverage Path Planning

Stanley Brown and Steven L. Waslander
University of Waterloo

2017. 05. 18

Presented by Suzi Kim

2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

The Constriction Decomposition Method for Coverage Path Planning

Stanley Brown* and Steven L. Waslander†

***Abstract*— The task of coverage path planning in 2D indoor and outdoor environments is classified as a NP-hard problem, and has been an active research topic for over 30 years. We derive a novel, exact cellular decomposition method called the Constriction Decomposition Method (CDM) and apply it to complex indoor environments. The CDM rapidly identifies constriction points in the environment and decomposes the environment into easily traversable cells by exploiting the geometric information contained in the environment's straight skeleton. Several heuristic path planning methods that find paths that completely cover each cell are explored. We apply our method on a complex indoor office-like environment and compare our results to existing methods.**

I. INTRODUCTION

Coverage path planning (CPP) is the task of determining an optimal set of paths that allow a robot or agent to transverse all free space in an environment. CPP has numerous applications for tasks that require the entirety of free space in an environment to be covered, such as crop harvesting, inspection, and mapping. The focus of this paper is the derivation of a CPP algorithm that produces coverage paths tailored to operation in complex human environments with corridors, rooms and islands, by identifying constriction points in known maps at which to divide the environment in a logical manner.

The CPP problem has been an active research topic since at least the 1980s [1] and several main strategies for approaching the problem have emerged. In many cases, the environment is broken down, either explicitly or implicitly, into cells or sub-regions to simplify the generation of coverage paths. If the cells are non-intersecting and the union of all cells fill the entire environment, the decomposition is called an exact cellular decomposition [2]. Examples of exact cellular decomposition include the Boustrophedon decomposition [3], Morse cell decomposition [2], minimum sum of angles (MSA) decomposition [4] or greedy convex polygon decomposition [5]. The Spiral Spanning Tree Coverage (Spiral-STC) method by [6] is an example of an approximate cellular decomposition, which is also simultaneously classified as a grid based method.

CPP algorithms can be classified as either offline or online methods [1]. Online CPP algorithms assume that limited or no prior knowledge of the environment exists and usually rely on a set of heuristic methods to ensure complete coverage [1]. Examples online CPP algorithms can be found in works of [5] and [7].

* M.A.Sc. Candidate, Mechanical and Mechatronics Engineering, University of Waterloo; s52brown@uwaterloo.ca
† Assistant Professor, Mechanical and Mechatronics Engineering, University of Waterloo; stevenw@uwaterloo.ca

In this paper, we propose an offline method that generates a coverage path plan using a novel, exact cellular decomposition called the constriction decomposition method (CDM). Our method works by decomposing a pre-mapped environment into a set of sells cells based on constriction points, which are defined as areas where the narrowest point is less than the two or more neighbouring areas. For indoor environments, a hallway between two large rooms can be thought of as a constriction point and by decomposing the environment into cells based on these points, an intuitive, room-based decomposition results. It is also shown that be resulting cells can be completely covered using a set of simple contour following paths, followed by a series of back and forth paths that are aligned with the longest edge of the cell.

II. RELATED WORK

One of the earliest exact cellular decomposition methods developed is the trapezoidal decomposition method which breaks polygonal environment down into cells by inserting a set of edges up and down from each obstacle vertex to the bounding polygon [8]. Every cell that is generated in this fashion is trapezoidal, convex and can be covered in a series of back and forth motions that are parallel with the direction of the inserted edges. While the trapezoidal decomposition method is shown to be algorithmically complete, it typically generates a number of cells that could be merged and still be covered using the same coverage pattern. This idea is formalized in the derivation of the Boustrophedon Decomposition [3].

The word boustrophedon comes from ancient Greek and literally means "the way of the ox" [3]. In the Boustrophedon Decomposition, a set of cells are generated assuming that the robot covers a space in a series of back and forth motions similar to the way farmers plow or work a field. By running a scan line that is parallel to the direction of the back and forth motions the robot will take when covering an environment, and inserting an edge whenever an edge can be drawn both above and below the vertex of an obstacle, a set of cells is created. These cells can always be covered using a boustrophedon like coverage pattern and by visiting all the cells the method is shown to be complete [3].

In later work, Choset [2] demonstrated that the Boustrophedon Decomposition can be generalized by proposing a cellular decomposition based on the critical points of Morse functions. The Morse Decomposition can be generalized to any n-dimensional space [2] and also allows for different coverage patterns to be used such as spiral, circular, or diamond patterns. In the case of the Boustrophedon Decomposition, its coverage pattern can be represented by the

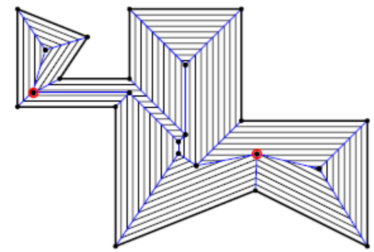


Fig. 1: An example straight skeleton in a non-convex polygon along with offset polygons located at $t = \{1, 2, ..., 10\}$. Skeleton arcs are highlighted in with thin blue lines, skeleton nodes are denoted by black dots and nodes that correspond to split events are circled in red.

to move along the angular bisector of adjacent edges, forming edges in the straight skeleton. This process continues as long as the boundary does not change topologically.

During the shrinking process, there are two possible topological changes that can occur: an edge event and a split event [13]. An edge event occurs when an edge, e_i, shrinks to a length of zero, which causes its neighbouring edges to become adjacent. A split event occurs when a reflex vertex of the wavefront collides with an edge and causes it to split into two new edges, which in turn split the wavefront into two separate polygons. The shrinking process then still continues in all polygons until all edges shrink to a length of zero. The bisector line segments traced out by the vertices of the edges during the entire process are called the skeleton edges, and the bisector arc start and end points are called the nodes of $S(F)$. A node is said to be a contour node if it corresponds to a vertex in V and a skeleton node if it does not.

The shrinking process also gives rise to a hierarchy of nested polygons, a subset of which are shown in Figure 1 for an example polygonal environment along with the skeleton arcs, edge and split nodes. In Figure 1, it can be noted that the location of a split node gives rise to an additional polygon at the corresponding time steps. For the example polygon, the first split occurs at $t = 3.6$ and second split occurs at $t = 5.8$.

The process of shrinking a polygon for some time t at speed, ω, can be related to the task of creating a set of contour following paths. If an agent with a width, w, performing a coverage operation was to follow the contours of the polygonal environment m times, the polygon introduced for the m^{th} pass is the same as the polygon created by the shrinking process detailed above at a time $t = w \cdot m$, and indeed, $\omega = w$ in this case. Therefore, the coverage problem time that each node in $S(F)$ was created corresponds to the number of times an agent can circle the free space of an environment without overlapping previous paths.

The location and time where the split nodes are created are of the greatest interest to our work as they encode the constriction points of an environment. Whenever a split event occurs, the resulting polygon of the wavefront is split into two separate polygons which means that the nodes on either side of the split node are created at a later time. Therefore, if the polygon is decomposed based on the split nodes in its straight skeleton, it is possible to create a set of cells that have no split nodes in their straight skeletons. That is, for such polygons, it will always be possible generate contour following paths that spiral inwards without crossing over a previous path, as described below. We term a decomposition based on this method the Constriction Decomposition Method (CDM).

B. The Constriction Decomposition (CDM)

Let N denote the set of all nodes, n_i, in the straight skeleton of the free space, $S(F)$, such that $n_i \in N$. Each node, n_i, is defined by its location in F, and has an associated time of creation, t_n, node type, $\tau_i = \{\tau^k, \tau^t\}$, (either skeleton or contour), event type, $\nu_i = \{\nu^s, \nu^e\}$, (split or edge) and a set of neighbouring nodes, M_i, connected by an edge in the straight skeleton graph. Given any node in the graph, its neighbours and their corresponding properties can be determined using graph traversal methods.

One method to find a set of decomposition edges E' such that F is decomposed into a set of cells C based on N is to simply search the skeleton of F, $S(F)$, for any split nodes. When a split node, n_{split}, is found, a new vertex, v_{new}, is inserted into the edge $e_{n,m}$ impacted by the angular bisector of v_i such that the length of e_{new} is minimized. In the case where the nearest point on the edge is equal to one of its vertices, e_{new}, is inserted between that vertex and v_i and no additional vertex is added to F. The insertion of the edge e_{new} results in a new cell $c_i \in F$ being created and also removes a split event from the skeleton of the remaining, non-decomposed part of $F' = F \backslash c_i$. This process is repeated until no split nodes exist and F is completely decomposed into a set of cells C. The CDM process is summarized algorithmically below and applied to the polygon in Figure 2.

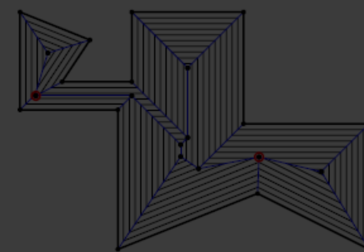
Let $\theta : N \rightarrow \{\tau_i^k, \tau_i^t\}$ be function that takes a node $n_i \in N$ and returns its type, either contour or skeleton, and let $\psi : N \rightarrow \{\nu_i^s, \nu_i^e\}$ be a function that takes a node and returns its event type. Defining $\eta(v_i, e_{n,m})$ to be a function that takes a vertex, $v_i \in V$, and an edge, $e_{n,m} \in E$, and returns a virtual vertex, $v*$ on the edge, $e_{n,m}$ such that the minimum distance between v_i and v^* is minimized. Then the CDM process can be described as follows in Algorithm 1.

Errors in Equations

Errors in Algorithm

Poor Explanation

Lack in Experiments



THEN, WHY TO PRESENT THIS PAPER?

1. No recent paper dealing with the topic “Coverage Path Planning on 2D Map”

Google

coverage path planning

학술검색 검색결과 약 20,500개 (0.03초) 내 서지정보

학술자료

내 서재

모든 날짜

2017 년부터

2016 년부터

2013 년부터

기간 설정...

2015 —

관련도별 정렬

날짜별 정렬

모든 언어

한국어 웹

특허 포함

서지정보 포함

도움말: 한국어 검색결과만 보기. 학술 검색 설정.에서 검색 언어를 선택할 수 있습니다.

Three-dimensional **coverage path planning** via viewpoint resampling and tour optimization for aerial robots

A Bircher, M Kamel, K Alexis, M Burri... - Autonomous ..., 2016 - Springer

Abstract This paper presents a new algorithm for three-dimensional **coverage path planning** for autonomous structural inspection operations using aerial robots. The proposed approach is capable of computing short inspection paths via an alternating two-step optimization ...

17회 인용 관련 학술자료 전체 5개의 버전 인용 저장

Coverage Path Planning with Real-time Replanning and Surface Reconstruction for Inspection of Three-dimensional Underwater Structures using Autonomous ...

E Galceran, R Campos, N Palomeras ... - Journal of Field ..., 2015 - Wiley Online Library

... a novel coverage path planning method for inspecting complex structures on the ocean floor using an autonomous underwater vehicle (AUV). The method initially uses a 2.5-dimensional (2.5 D) prior bathymetric map to plan a nominal **coverage path** that

18회 인용 관련 학술자료 전체 5개의 버전 인용 저장

Energy-aware **coverage path planning** of UAVs

C Di Franco, G Buttazzo - Autonomous Robot Systems and ..., 2015 - ieeexplore.ieee.org

Abstract: **Coverage path planning** is the operation of finding a **path** that covers all the points of a specific area. Thanks to the recent advances of hardware technology, Unmanned Aerial Vehicles (UAVs) are starting to be used for photogrammetric sensing of large areas in

21회 인용 관련 학술자료 전체 7개의 버전 인용 저장

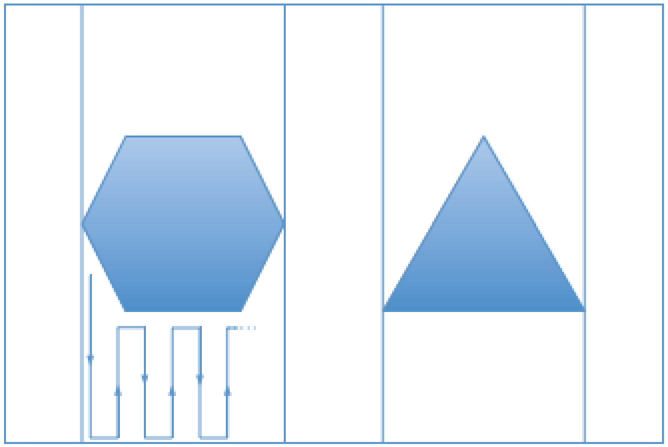
[인용] A new 3D **Coverage Path Planning** Approach for Agricultural Robots to Minimize Skip Areas between Swaths over Fields of Complex Terrain

I Hameed, A la Cour-Harbo - Robotics and Autonomous Systems, 2015

2. Coverage path planning with topological analysis is quite interesting.

Coverage Path Planning (CPP)

- Determining a path that passes over all points of an area while avoiding obstacles.
- Generally, using back and forth motion after cell decomposition.
- Ex) Vacuum cleaning robots, painter robots, autonomous underwater robots, lawn mowers, automated harvesters



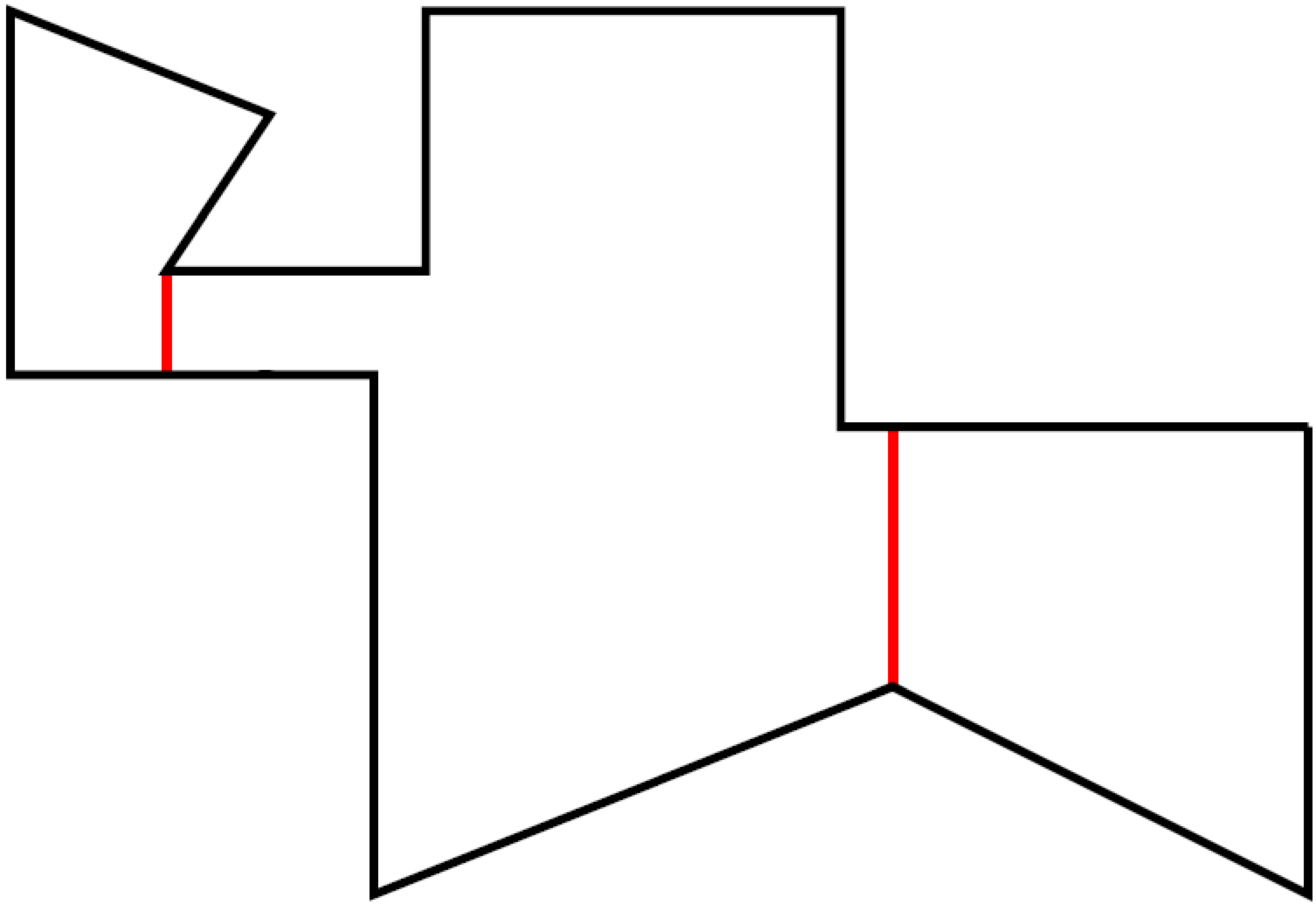
I hope…

CPP implies Space Efficiency

Maybe no direct relation.
At least weak relation…!

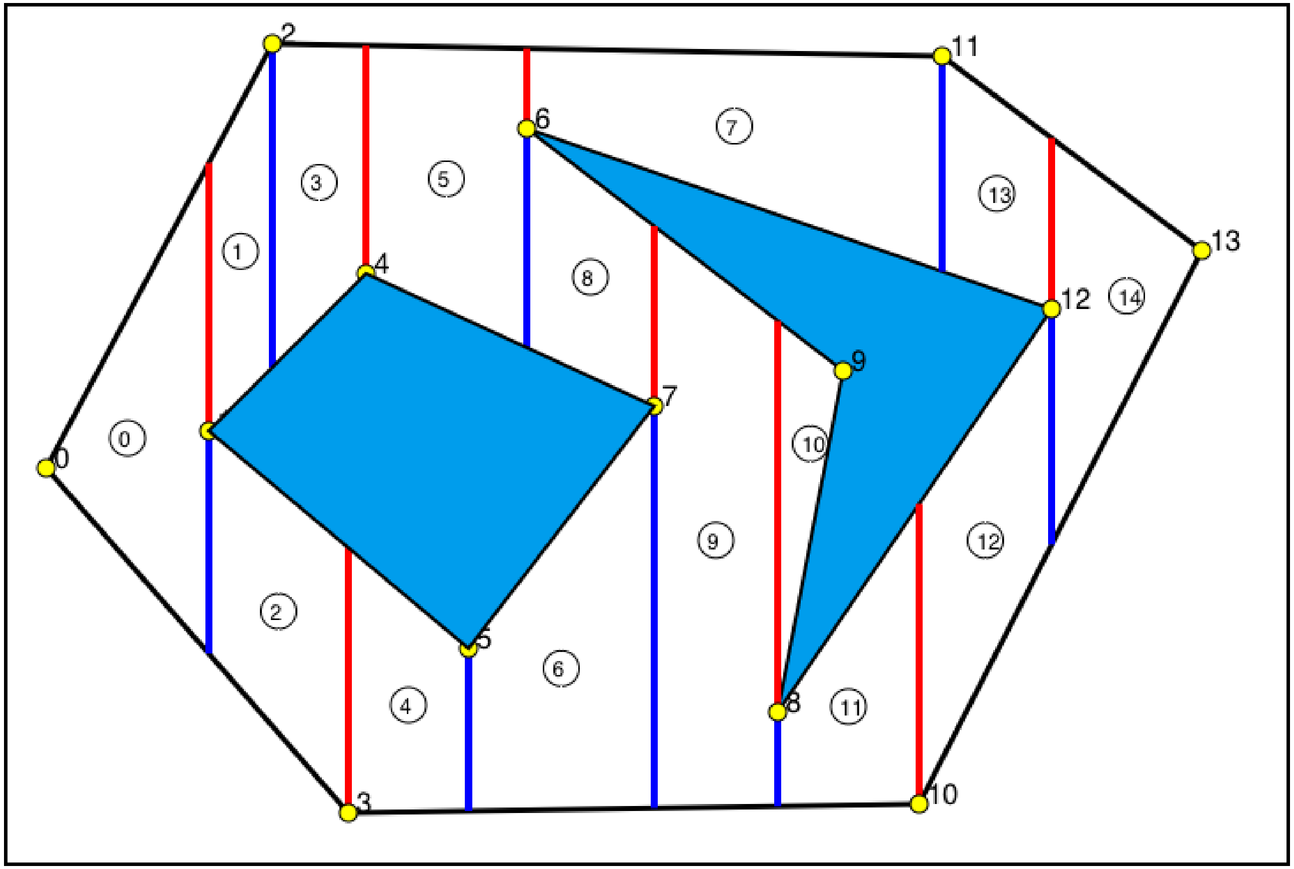
Some Critical Points Inside the Polygon Map

The Constriction Decomposition Method for Coverage Path Planning



[Previous Work]

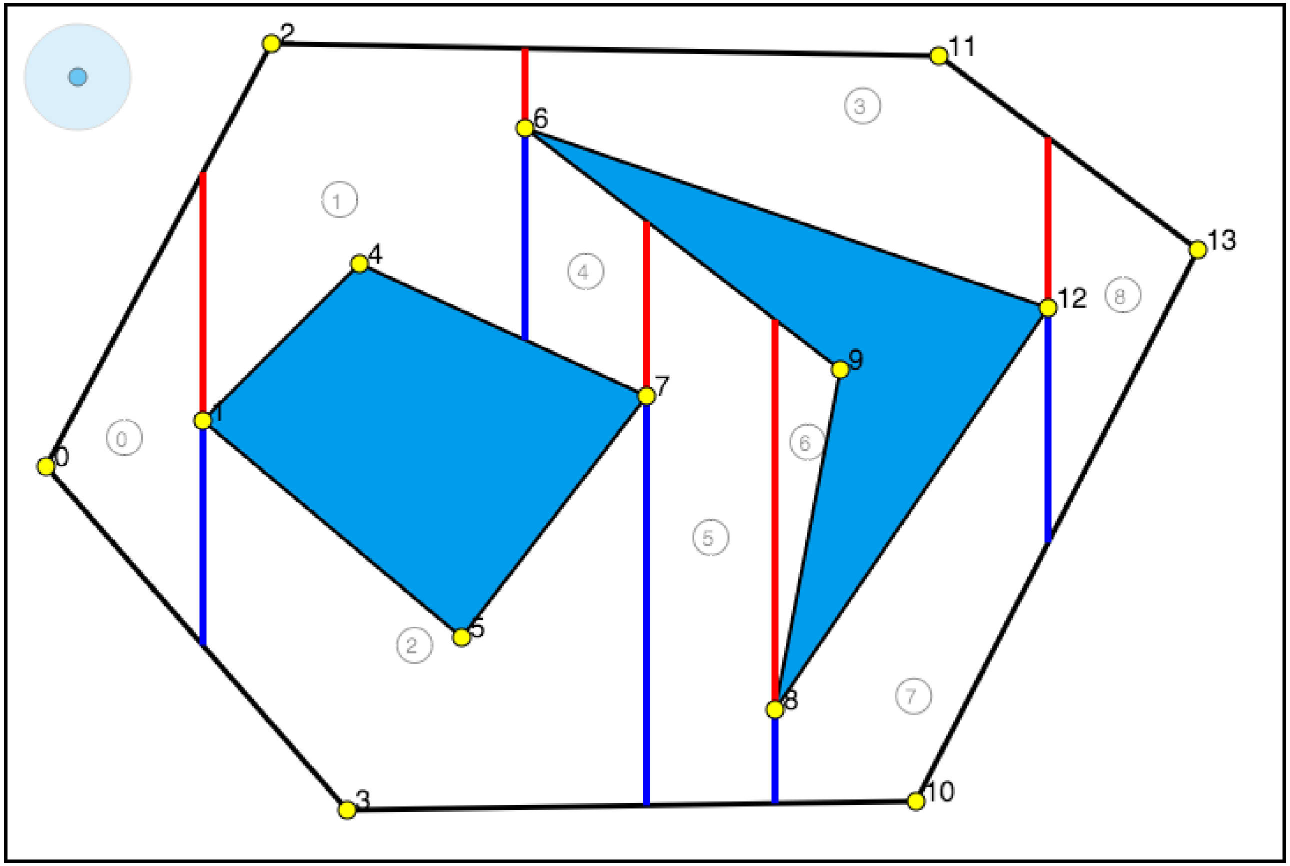
Trapezoidal Decomposition



http://user.ceng.metu.edu.tr/~akifakkus/courses/ceng786/hw3.html

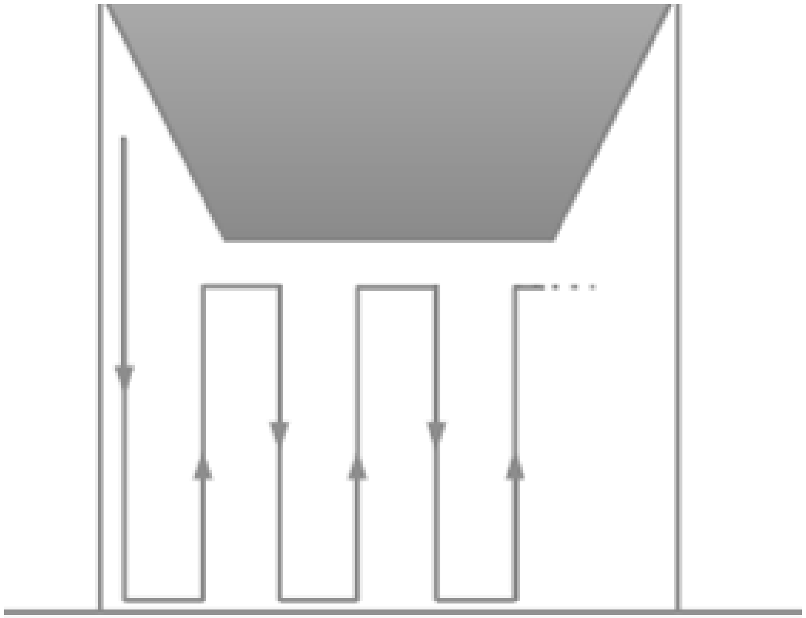
[Previous Work]

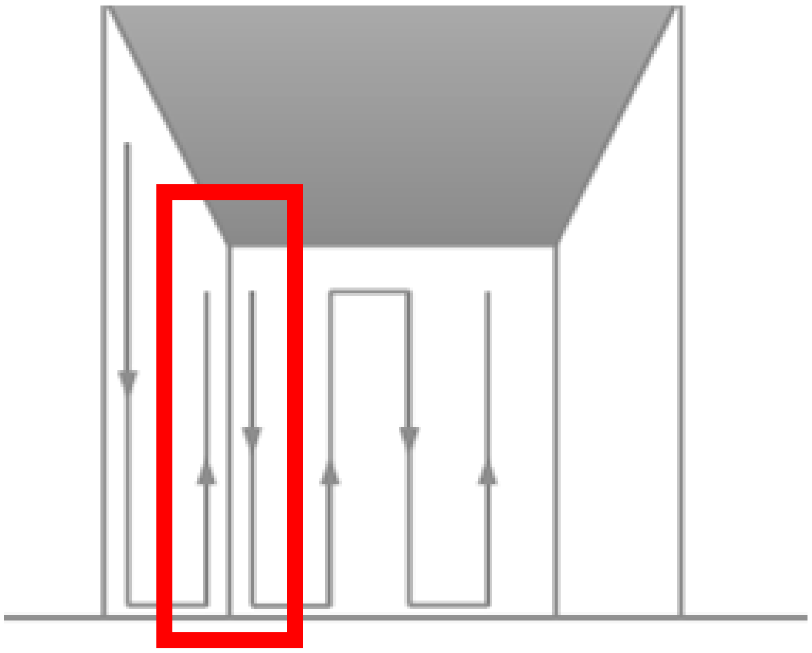
Boustrophedon Decomposition



http://user.ceng.metu.edu.tr/~akifakkus/courses/ceng786/hw3.html

Why Less Cells Are Better?





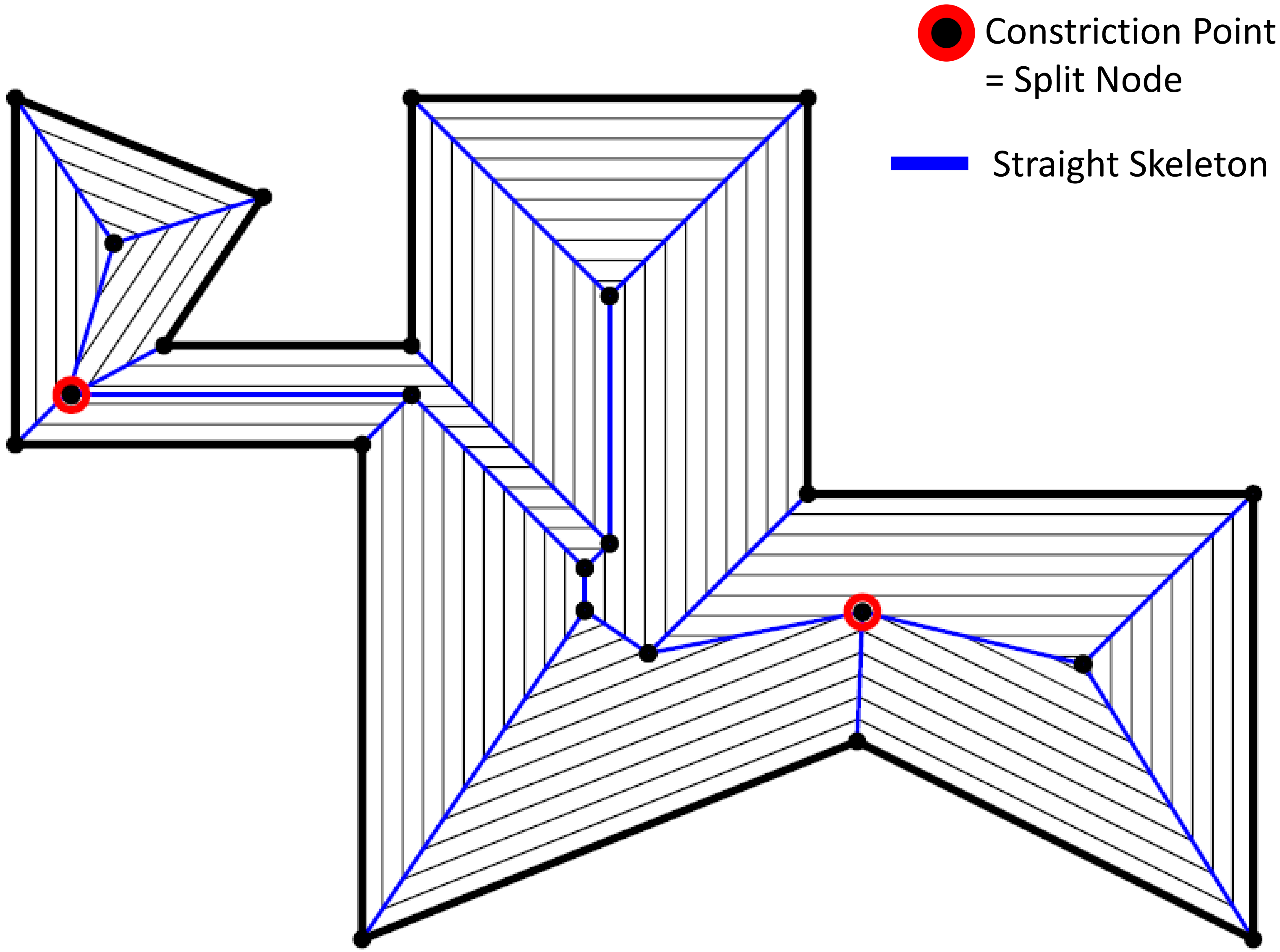
Galceran, Enric, and Marc Carreras. "A survey on coverage path planning for robotics." *Robotics and Autonomous Systems*, 2013

Constriction Decomposition Method

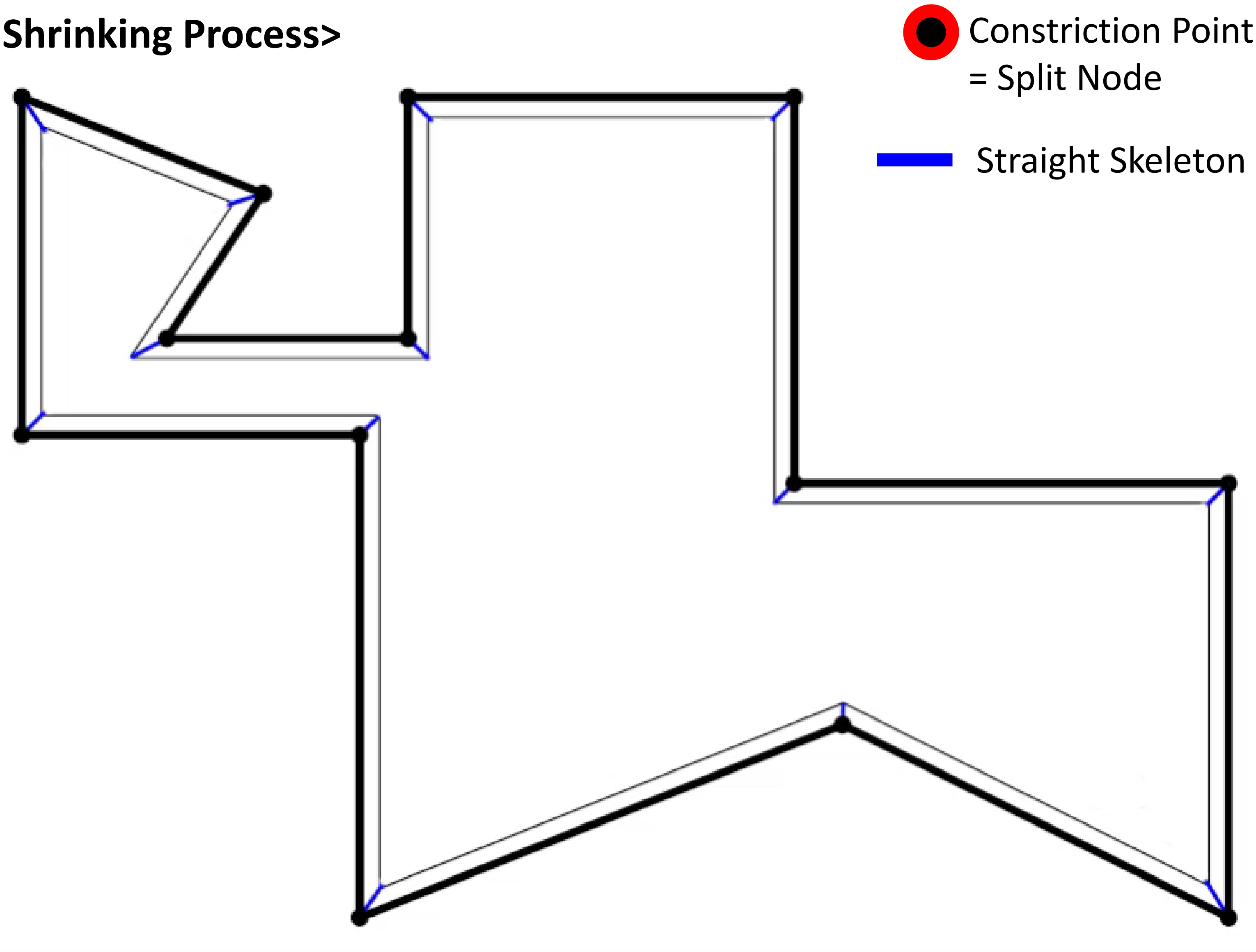
Constriction Decomposition Method

1. Generate a Straight Skeleton.

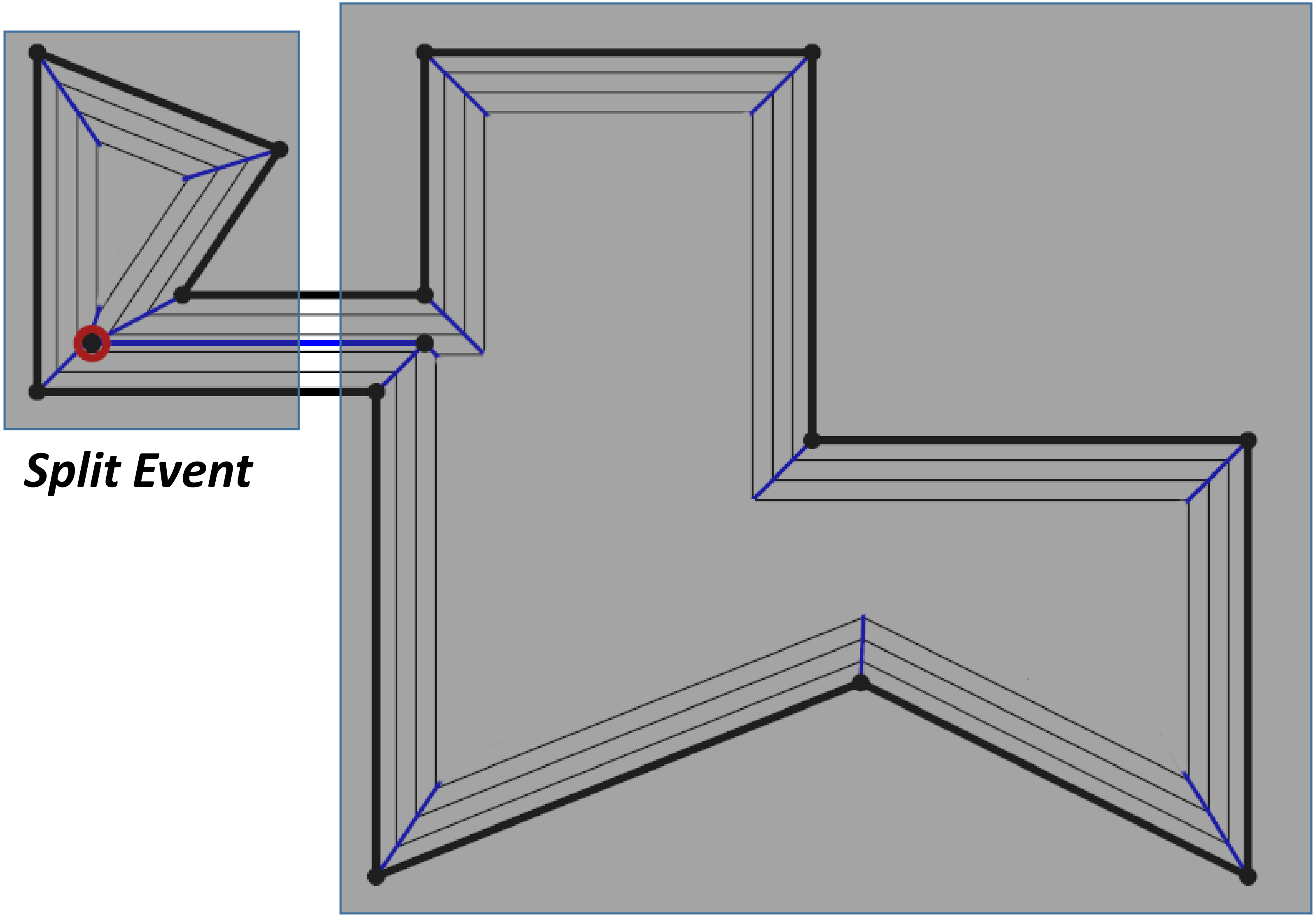
2. For all split nodes,
   Generate a separator from a neighbor node of split nodes on boundary.


Constriction Point
= Split Node
Straight Skeleton

<Shrinking Process>

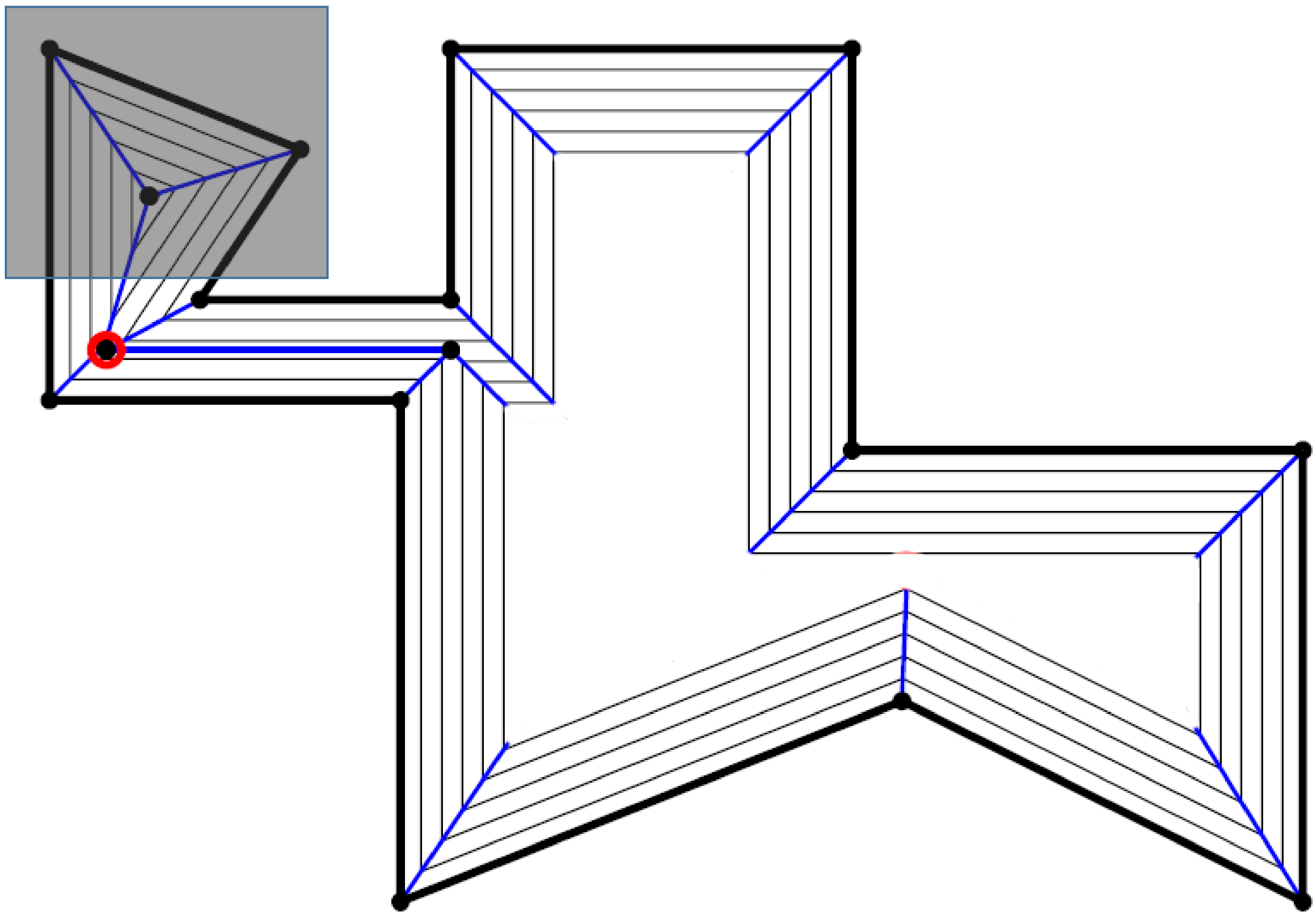


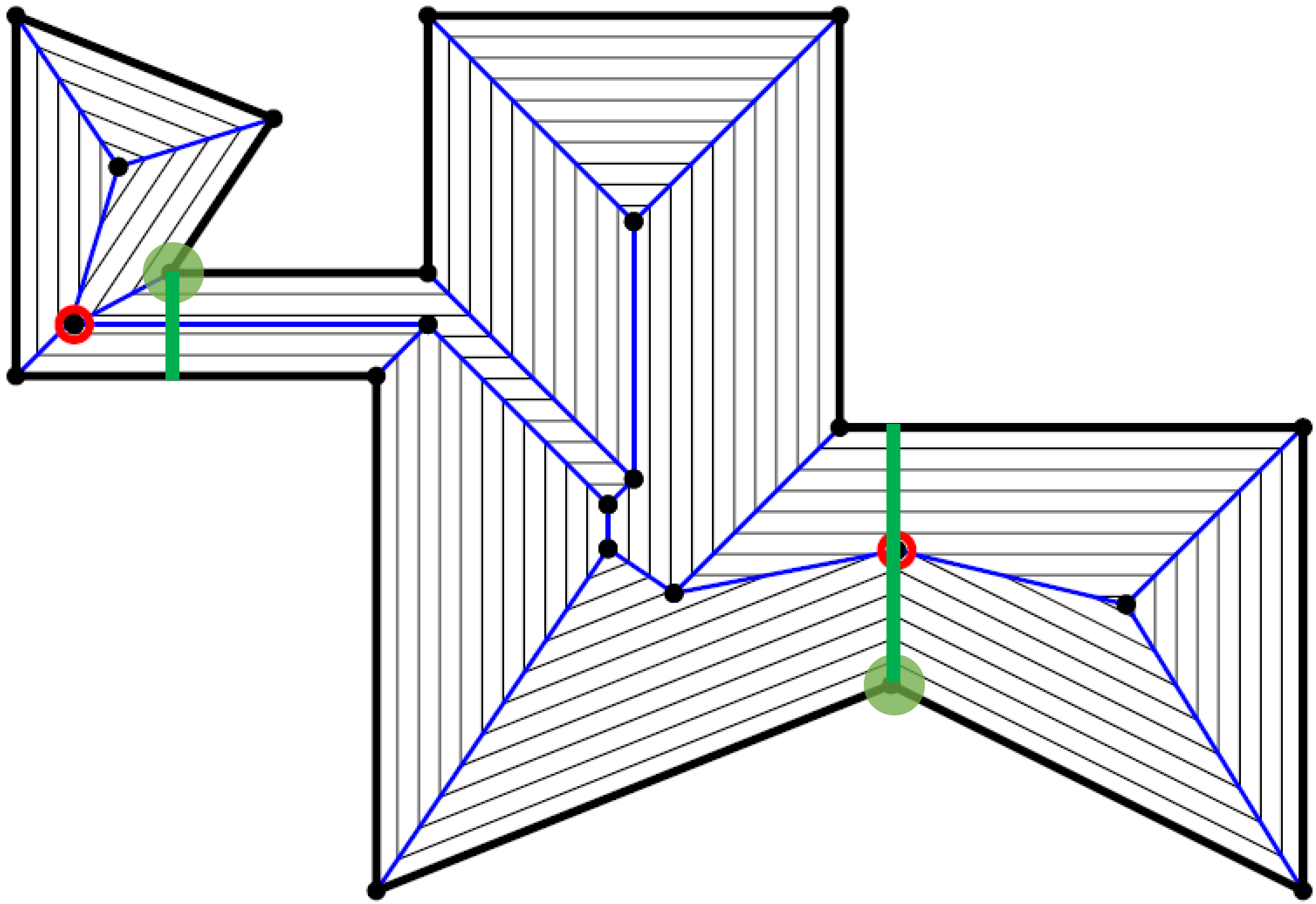
Aichholzer, Oswin, et al. "A novel type of skeleton for polygons.", 1996.
Felkel, Petr, and Stepan Obdrzalek. "Straight skeleton implementation.", 1998.

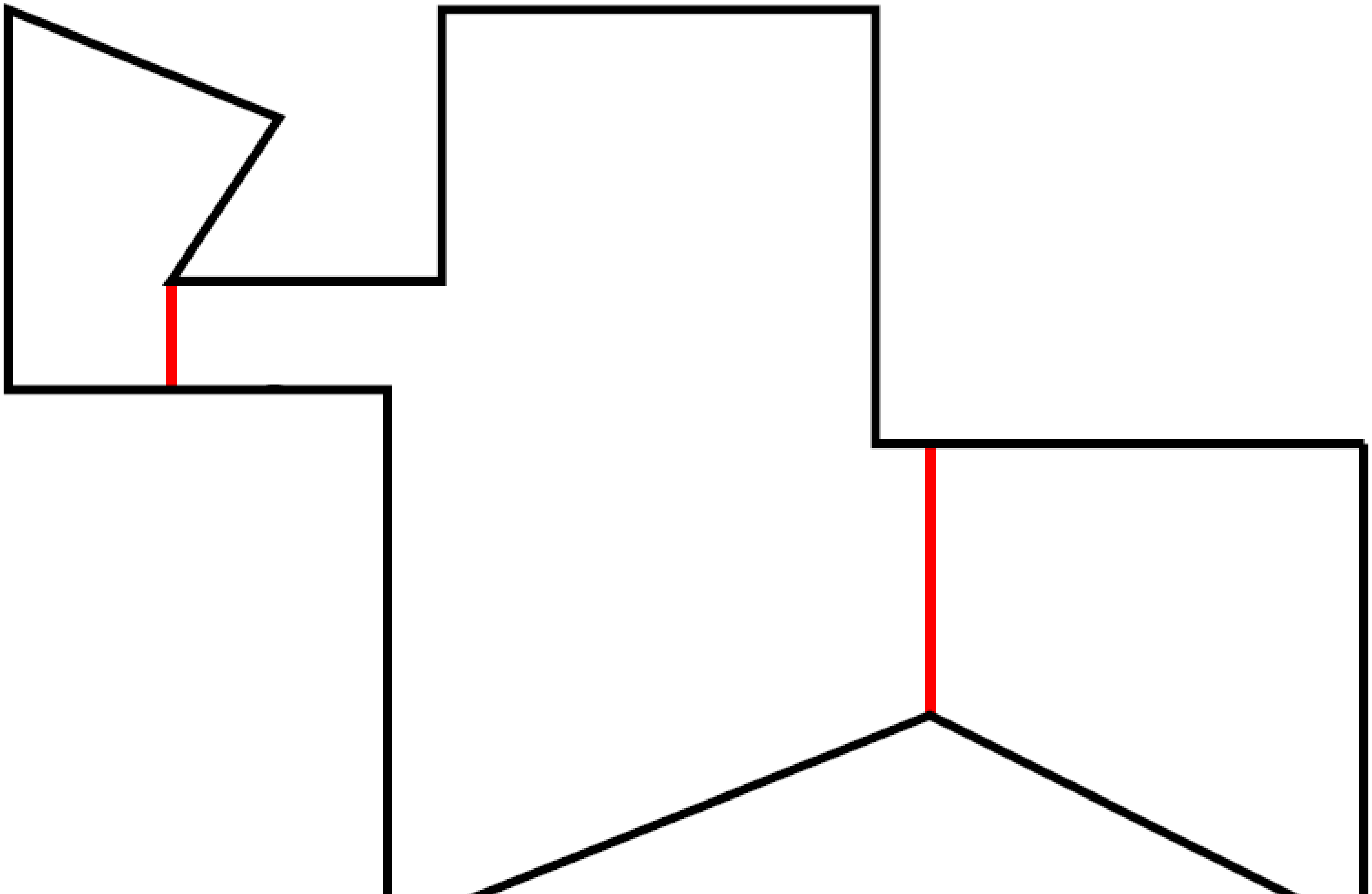


Split Event

Edge Event

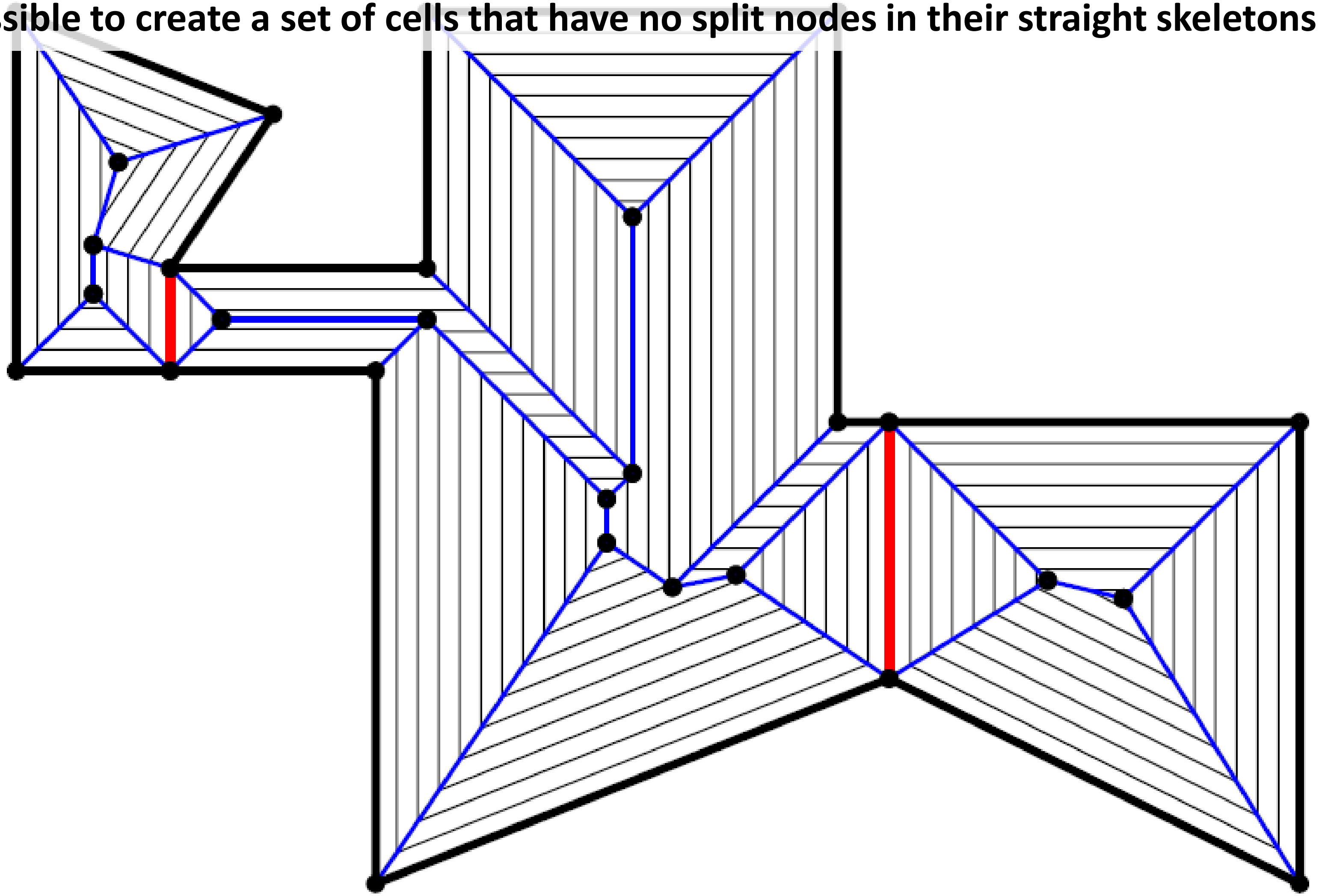


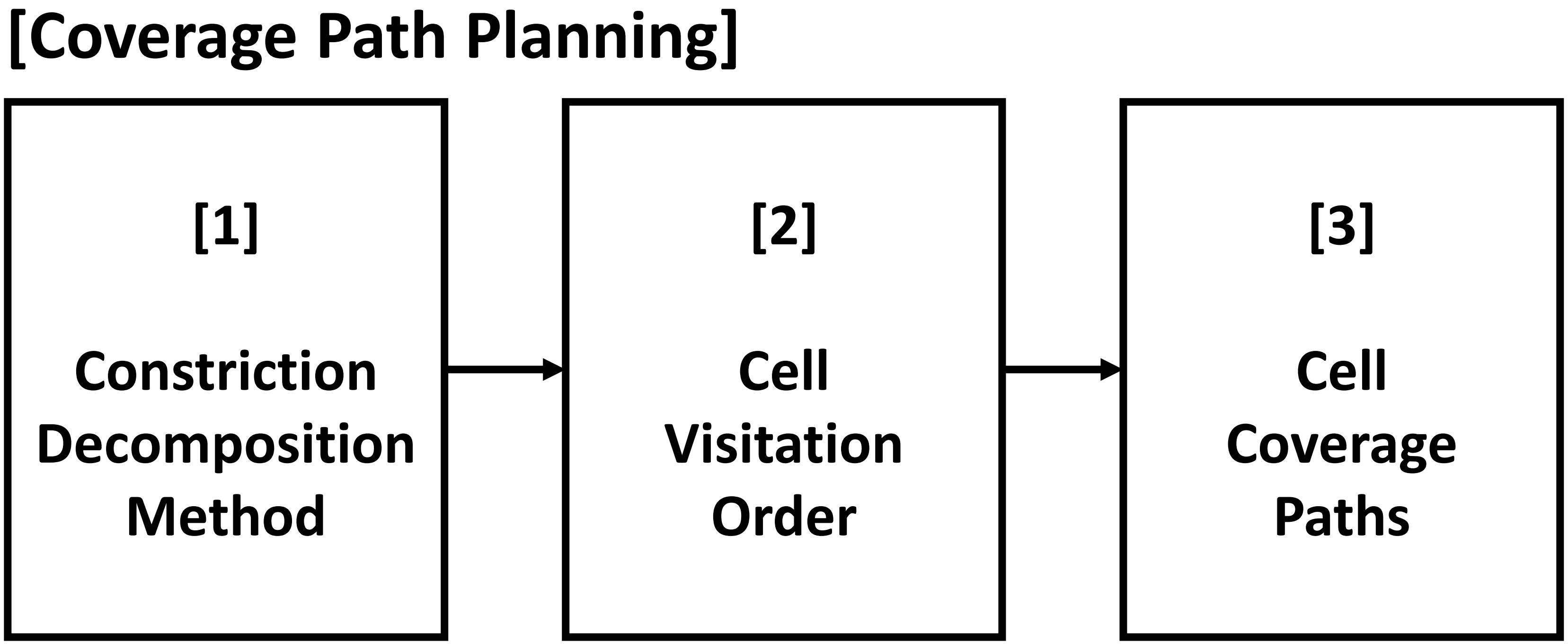




Can't Find Mathematical Background..

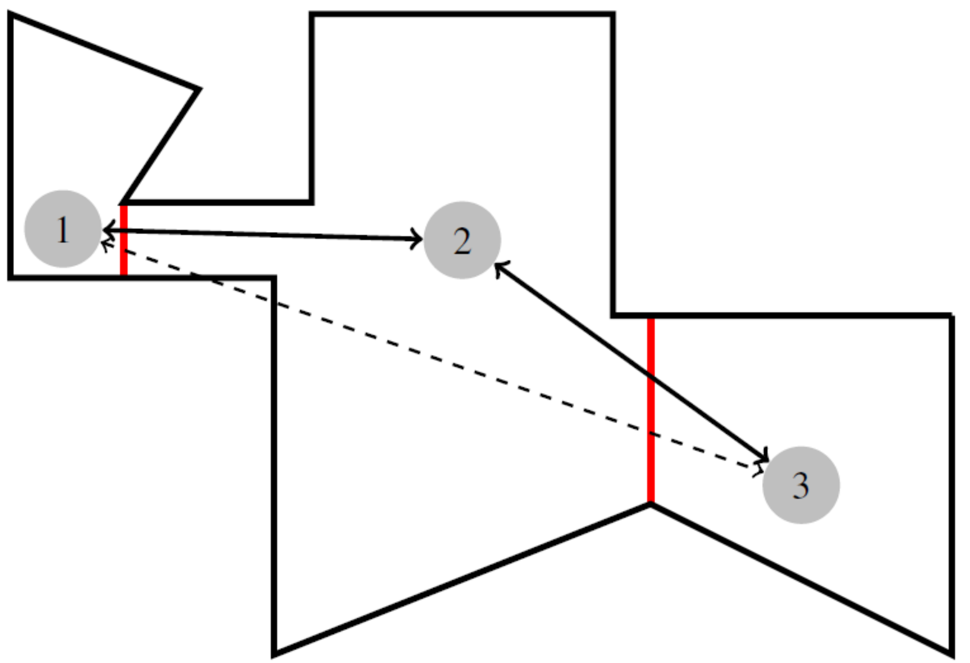
If the polygon is decomposed based on the split nodes in its straight skeleton,
It is possible to create a set of cells that have no split nodes in their straight skeletons.




[Coverage Path Planning]
[1]
Constriction Decomposition Method
[2]
Cell Visitation Order
[3]
Cell Coverage Paths

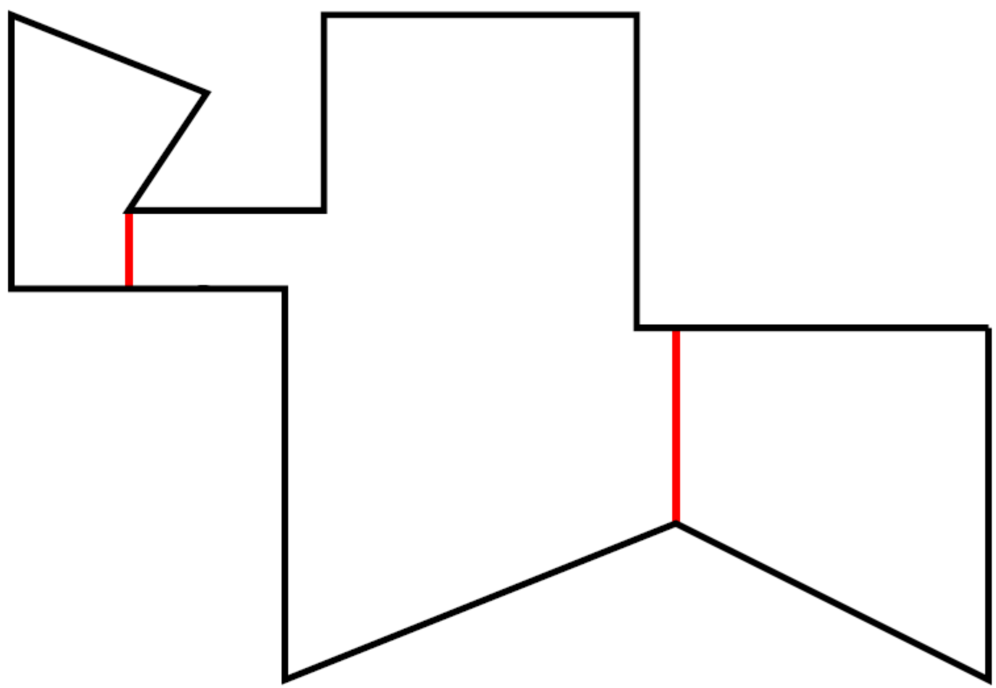
[2] Cell Visitation Order

- Should visit every cell.
- Generate an augmented adjacency graph.
- Solve using a *Heuristic TSP* algorithm.



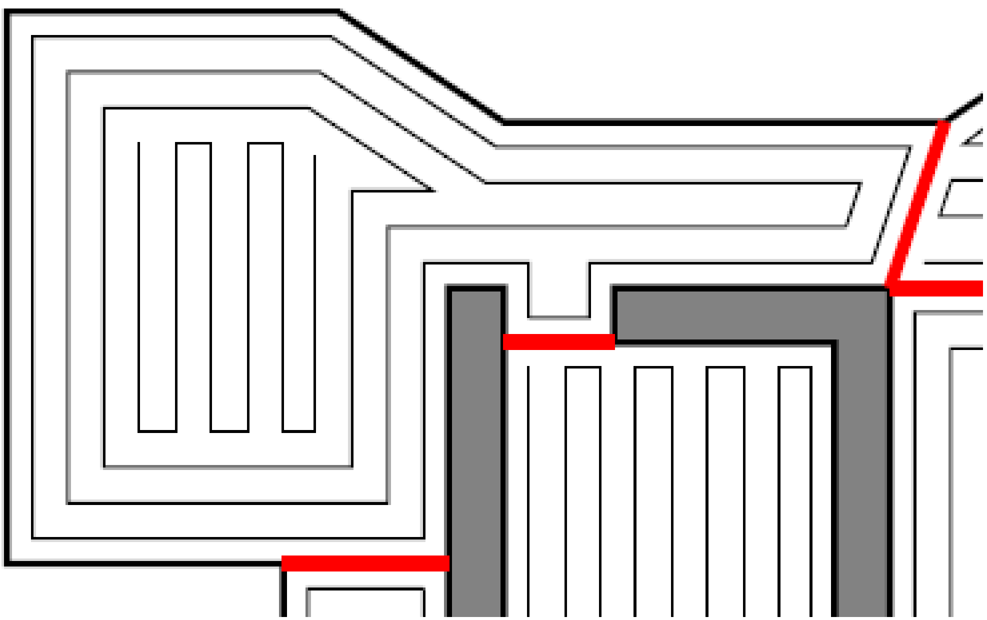
[3] Cell Coverage Paths

- Applying Boustrophedon Decomposition to non-convex polygon directly might lead to over decomposition depending on the incline of the scan line.



[3] Cell Coverage Paths

1. Spiraling until the remaining free space forms a convex shape.
2. Generating *Boustrophedon paths* for the remaining convex shape area.



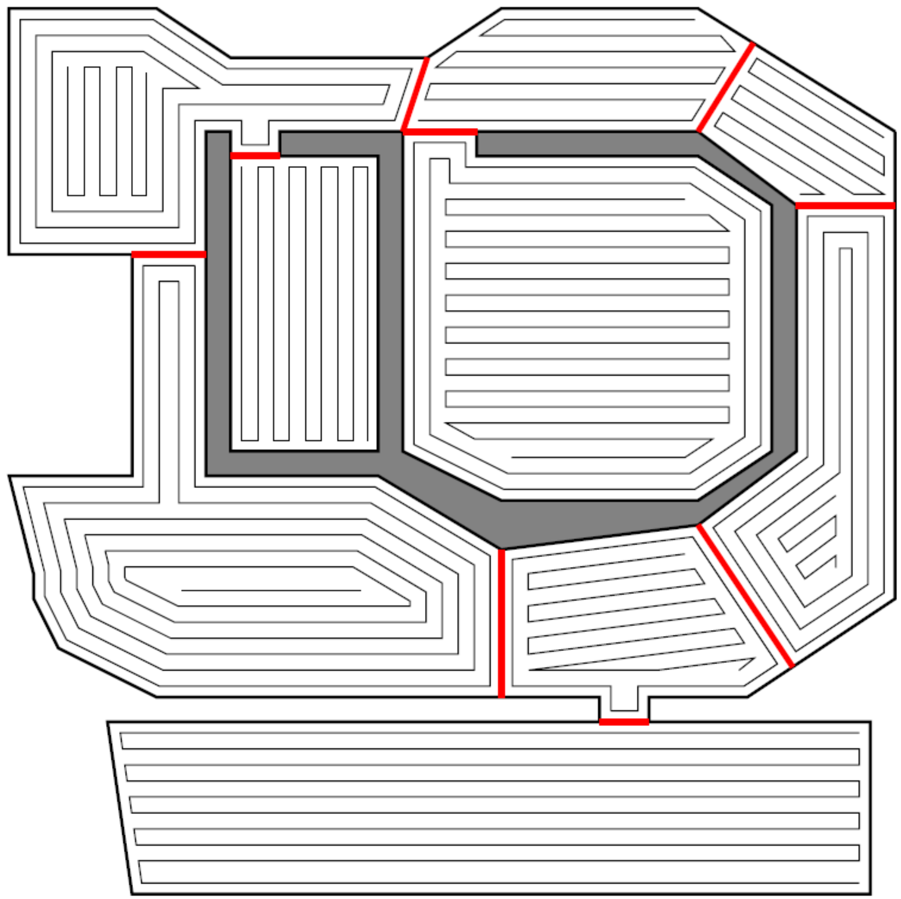
Results

Environment	BSD	CDM	Improvement
Non-convex	3	3	0%
Non-convex, with holes 1	14	9	36%
Non-convex, with holes 2	11	3	73%
Non-convex, with holes 3	9	5	44%
Floor Plan 1	169	53	68%
Floor Plan 2	158	65	58%
Floor Plan 3	172	66	62%

BSD: Boustrophedon Decomposition
CDM: Constriction Decomposition Method

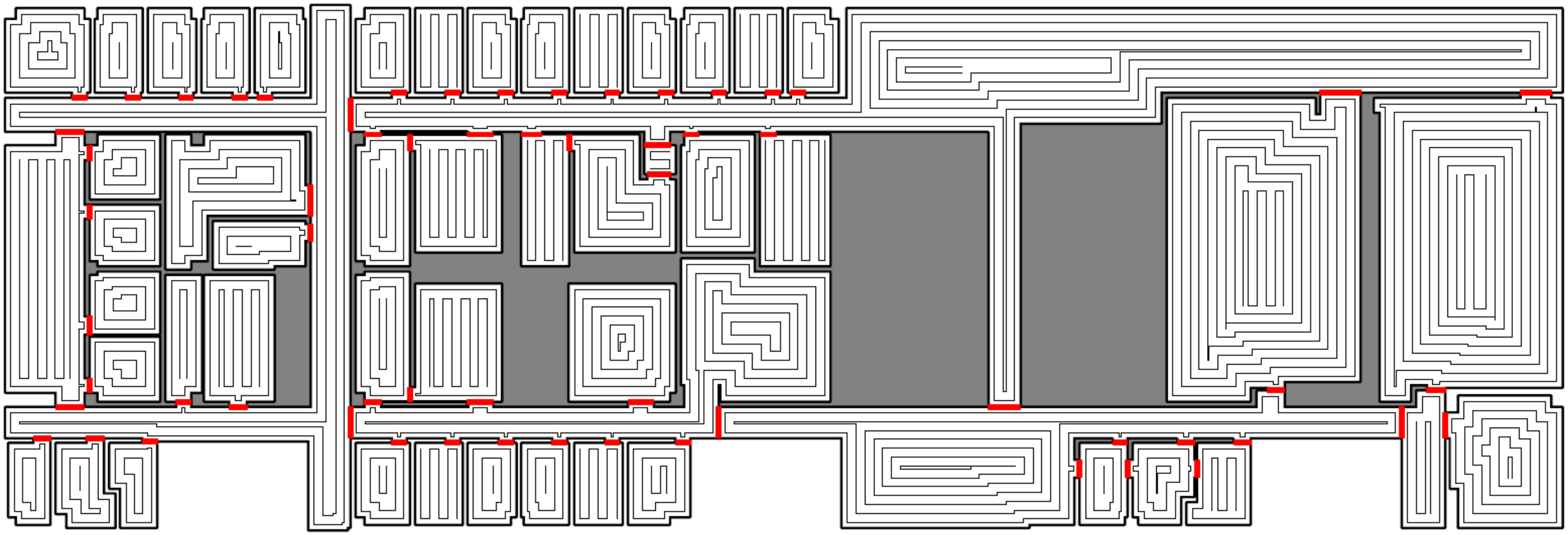
Results

Environment	BSD	CDM	Improvement
Non-convex, with holes 1	14	9	36%



Results

Environment	BSD	CDM	Improvement
Floor Plan 1	169	53	68%



Conclusion

- Strength in complex indoor environments containing lots of hallways and rooms.
- Doubt in '*Cell Coverage Paths*' whether it is really better than using either of spiraling or *Boustrophedon paths.*