Polygon Decomposition

J. Mark Keil

Department of Computer Science
University of Saskatchewan
Saskatoon, Sask., Canada, S7N 5A9

May 14, 1996

0.1 Introduction

Practitioners frequently use polygons to model objects in applications where geometry is important. In polygon decomposition we represent a polygon as the union of a number of simpler component parts. Polygon decomposition has many theoretical and practical applications and has received attention in several previous surveys [26, 68, 107, 128, 133].

Pattern Recognition is one area that uses polygon decomposition as a tool [41, 110, 111, 112, 133]. Pattern recognition techniques extract information from an object in order to describe, identify or classify it. An established strategy for recognising a general polygonal object is to decompose it into simpler components, then identify the components and their interrelationships and use this information to determine the shape of the object [41, 111].

Polygon decomposition is also useful in problems arising in VLSI artwork data processing. Layouts are represented as polygons, and one approach to preparation for electron-beam lithography is to decompose these polygon regions into fundamental figures [6, 42, 100, 102]. Polygon decomposition is also used in the process of dividing the routing region into channels [82].

In computational geometry, algorithms for problems on general polygons are often more complex than those for restricted types of polygons such as convex or star-shaped. The point inclusion problem is one example [114]. For other examples see [4] or [108]. A strategy for solving some of these types of problems on general polygons is to decompose the polygon into simple component parts, solve the problem on each component using a specialized algorithm, and then combine the partial solutions.

Other applications of polygon decomposition include data compression [92], database systems [90], image processing [97], and computer graphics [131].

Although much work has been done on decomposing polyhedra in three or higher dimensions [17, 10], we will restrict the scope of this survey to that of decomposing polygons in the plane.

Triangulation, the partitioning of the interior of a polygon into triangles, is a central problem in computational geometry. Many algorithms for polygons begin by triangulating the polygon. As early as 1978, Garey et al [45] provided an $O(nlogn)$ time algorithm, but no matching lower bound was known. The importance of the problem led to a significant amount of research [39, 132] on algorithms, culminating in Chazelle's linear time algorithm [27]. Although it certainly is an example of a polygon decomposition problem, the triangulation problem has taken on a life of its own and we will consider a systematic study of the triangulation problem, as well as related mesh generation work, to be outside the scope of this survey. For a good survey of mesh generation and optimal triangulation see [17].

There are a wide variety of types of component subpolygons that are useful for polygon decomposition. These include subpolygons that are convex, star-shaped, spiral or monotone, as well as fixed shapes such as squares, rectangles and trapezoids. Before proceeding further we provide definitions for some of the restricted types of polygons. A point x in a polygon P is visible from a point y in P, if the line segment joining x and y lies entirely inside P. We treat a polygon as a closed set, thus a visibility line may touch the boundary of P. A polygon P is *convex* if every pair of points in P are visible from each other. A polygon P is *star-shaped* if there exists at least one point x inside P from which the entire polygon is visible. The entire set of points in P from which P is visible is called the *kernel* of P. A *polygonal chain* in a polygon P is a sequence of consecutive vertices of P. A *spiral polygon* is a polygon whose boundary chain contains precisely one concave subchain. A polygonal chain is *monotone* with respect to a line l if the projections of the vertices in the chain on l occur in exactly the same order as the vertices in the chain. A polygon P is *monotone* if there exists a line l such that the boundary of P can be partitioned into two polygonal chains which are monotone with respect to the line l. See figure 1.

It is also useful to classify the type of polygon that is being decomposed. Polygons may be simply connected or they may contain holes. Holes are nonoverlapping "island" simple polygons, inside the main polygon. Some authors allow for degenerate holes such as line segments or points. The complexity of a decomposition problem usually increases if the polygon contains holes. A polygon is said to be *orthogonal* if all of its sides are either horizontal or vertical. Orthogonal polygons are relevant in many applications and in this survey special emphasis is placed on the decomposition of orthogonal polygons.

Polygon decompositions are also classified according to how the component parts interrelate. A decomposition is called a *partition*, if the component subpolygons do not overlap except at their boundaries. If generally overlapping pieces are allowed we call the decomposition a *cover*.

Decomposing a polygon into simpler components can be done with or without introducing additional vertices which are commonly called Steiner points. While the use of Steiner points makes subsequent processing of the decomposed polygon more complex, it also often allows the use of fewer component parts. See figure 2.

In a polygon with n vertices, at some N of the vertices the interior angle will be reflex (greater than $180°$). The number N of reflex vertices of a polygon can be much smaller than n and we analyze the complexity of decomposition algorithms with respect to both n and N. See figure 2.

In most applications we want a decomposition that is minimal in some sense. Some applications seek to decompose the polygon into the minimum number of some type of component. Other applications use a decomposition that minimizes the total length of the internal edges used to form the decomposition (minimum "ink"). Perhaps the earliest minimum "ink" result is due to Klincsek [69]. He uses dynamic programming to find the minimum "ink" triangulation of a polygon. His work was influential in that it inspired subsequent dynamic programming solutions to decomposition problems. As in the example of figure 6, a minimum edge length decomposition can be quite different from a minimum number decomposition for the same component type.

In the next section we will review the work that has been done concerning partitioning and covering general polygons. In section 3 we turn our attention to orthogonal polygons and consider the work done on decomposing orthogonal polygons.

0.2 Decomposing General Polygons

In this section we consider both partitioning and covering problems for general polygons.

0.2.1 Polygon Partitioning

When partitioning a polygon into simpler subpolygons, it is the application which determines the type of subpolygon to be used. Syntactic pattern recognition uses convex, spiral and star-shaped decompositions [41, 110, 112, 123, 9, 133]. VLSI applications use trapezoids [6]. In the rest of this section we will consider each of these types of subpolygons in turn.

Convex Subpolygons

When the polygon may contain holes, the problem of partitioning a polygon into the minimum number of convex components is NP-hard [79], either allowing or disallowing Steiner points. For polygons without holes much of the work done disallows Steiner points. A 1975 algorithm, due to Feng and Pavlidis [41], runs in $O(N^3 n)$ time, but does not generally yield a minimum decomposition. Schachter's 1978 [123] $O(nN)$ time partitioning algorithm also cannot guarantee a minimum number of components.

For polygons without holes disallowing Steiner points, several approximation algorithms provide results guaranteed to be close to optimum. In 1982 Chazelle [25] provides an $O(nlogn)$ time algorithm that finds a partition that contains fewer than $4\frac{1}{3}$ times the optimal number of components.

Later, Greene [55], and Hertel and Mehlhorn [61] provide $O(nlogn)$ time algorithms that find a partition that contains less than or equal to four times the optimal number of components. Note that, for polygons without degenerate holes, any convex partition that does not contain unnecessary edges will be within four times of the optimal sized partition. This is true as each added edge can eliminate at most two reflex vertices and each reflex vertex requires at most two edges to eliminate it in any convex partition that does not contain unnecessary edges.

The year 1983 saw the achievement of algorithms obtaining the optimal number of convex components. When disallowing Steiner points, Greene [55] developed an $O(N^2n^2)$ time algorithm for partitioning a polygon into the minimum number of convex components. Independently, Keil [64, 65] developed an $O(N^2nlogn)$ time algorithm for the problem. This result employs a general technique for reducing the size of the state space in a dynamic programming formulation.

Allowing Steiner points makes the problem quite different. There are an infinite number of possible locations for Steiner points. Nevertheless, as early as 1979 Chazelle and Dobkin [24, 28, 29] developed an $O(n + N^3)$ time algorithm for the problem of partitioning a polygon into the minimum number of convex components. They define an X_k-pattern to be a particular interconnection of k reflex vertices which removes all the reflex angles at these vertices and creates no new reflex angles. See figure 3. They achieve their algorithm by further developing this idea and using a dynamic programming framework.

Dobkin et al [37] show how to extend the existing algorithms for decomposing a polygon without holes into convex components to optimally partition a splinegon into convex components (with or without Steiner points). A splinegon is a polygon where edges have been replaced by "well behaved curves" [37].

Partitioning a polygon with holes into convex components remains hard under the minimum edge length criterion. With Steiner points Lingas et al [82] show the problem is NP-hard. Disallowing Steiner points, Keil [64] shows the problem is NP-complete.

Allowing Steiner points, Levcopoulos and Lingas develop approximation algorithms for the problem [76]. For polygons without holes, they have an $O(nlogn)$ time algorithm that yields a solution of size $O(plogN)$, where p is the length of the perimeter of the polygon. For polygons with holes, they have an $O(nlogn)$ time algorithm that produces a convex partition of size $O((b + m)logN)$, where b is the total length of the boundary of the polygon and the holes and m is the minimum length of its convex partition. No optimal algorithms for the problem are known when Steiner points are allowed.

For a convex polygon with point holes, without the use of Steiner points, Plaisted and Hong [113] give a polynomial time algorithm for partitioning into convex subpolygons, such that the total edge length is within 12 times the minimum amount required. For this problem Levcopoulos and Krznaric [75] give a greedy type $O(nlogn)$ time algorithm that yields a solution that is also within a constant factor of optimal.

The year 1983 also saw the achievement of optimal algorithms under the minimum edge length criteria. For polygons without holes, disallowing Steiner points, Keil [64] develops an $O(N^2n^2logn)$ time dynamic programming algorithm for the problem of partitioning a polygon into convex subpolygons while minimizing the total internal edge length. Independently, Greene noticed that his algorithm for the convex minimum number problem [55] can be adapted to yield an $O(N^2n^2)$ time algorithm for the convex minimum edge length problem.

Spiral Subpolygons

Recall that a spiral polygon is a simple polygon whose boundary chain contains precisely one concave subchain. Keil [64] shows that the problem of partitioning a polygon with holes into the minimum number of spiral components is NP-complete, when Steiner points are disallowed. For polygons without holes, again disallowing Steiner points, Feng and Pavlidis [41] provide a polynomial time algorithm for the problem that does not generally yield the minimum number of components. Keil [65] provides an $O(n^3logn)$ time algorithm to partition a polygon without holes, disallowing Steiner points, into the minimum number of spiral components. He also provides an $O(n^4logn)$ time algorithm for the same problem under the minimum edge length optimality criterion [64]. No results are known concerning partitioning polygons into spiral components if Steiner points are allowed.

Star-shaped Subpolygons

Steiner points are disallowed in most of the known results concerning star-shaped partitioning. Again we see the hardness of decomposing a polygon with holes as Keil [64] shows that the problem of partitioning a polygon with holes into the minimum number of star-shaped components is NP-complete. In 1981, for polygons without holes, Avis and Toussaint [9] give an $O(nlogn)$ time algorithm that partitions a polygon into at most $\frac{n}{3}$ star-shaped components. This algorithm does not generally yield a minimum partition. In 1984 Aggarwal and Chazelle [2] are able to partition a polygon into $\frac{n}{3}$ components in $O(n)$ time.

In order to achieve a partition into the minimum number of star-shaped components, in 1983 Keil employs dynamic programming to develop an $O(n^5 N^2 logn)$ time algorithm [65]. The idea is to extend the solutions for small subpolygons into solutions for larger subpolygons. In general however, there can be an exponential number of minimum star-shaped partitions of a subpolygon. Furthermore, there are situations where no minimum partition of a subpolygon can be extended into a global minimum partition. The solution is to introduce pseudo star-shaped polygons. A pseudo star-shaped subpolygon has the property that there exists a point x in the polygon, but not in the subpolygon, so that every point in the subpolygon can be seen from x. The algorithm proceeds by keeping one star or pseudo star-shaped minimum partition of each of a number of equivalence classes of partitions at each subpolygon.

Shapira and Rappoport [125] make use of a form of star-shaped partition in a new method for the computer animation task of shape blending. They seek a partition into the minimum number of star-shaped components, each of whose kernels contains a vertex of the polygon. When such a partition exists, they compute it using a restriction of Keil's algorithm [64]. Since such a partition does not always exist, they also provide a heuristic which allows Steiner points.

For the problem of partitioning a polygon into star-shaped components while minimizing the total internal edge length, Keil [64] provides an $O(N^2 n^5 logn)$ time algorithm.

Monotone Subpolygons

Recall that a polygon P is monotone if there exists a line, l, such that the boundary of P can be partitioned into two polygonal chains, each of which is monotone with respect to l. For a polygon with holes, disallowing Steiner points, Keil [64] shows that the problem of partitioning a polygon into the minimum number of monotone subpolygons is NP-complete. For a polygon without holes, Keil [64] develops an $O(Nn^4)$ time algorithm for the problem. The algorithm relies on the fact that there are only a polynomial number of preferred directions with respect to which a subpolygon can be monotone. If a minimum partition is not important Garey et al [45] can provide an $O(nlogn)$ time algorithm.

If all of the subpolygons in a partition are monotone with respect to the same line then the partition is a decomposition into uniformly monotone components. Liu and Ntafos [88] give algorithms for partitioning a polygon without holes into the minimum number of uniformly monotone subpolygons. They give an $O(nN^3 + N^2 nlogn + N^5)$ time algorithm that does not use Steiner points, and an $O(N^3 nlogn + N^5)$ time algorithm that does allow Steiner points.

For the problem of partitioning a polygon into monotone components while minimizing the total internal edge length, Keil [64] gives an $O(Nn^4)$ time algorithm.

Other Subpolygons

The problem of partitioning a polygonal region into the minimum number of trapezoids, with two horizontal sides, arises in VLSI artwork processing systems [6]. A triangle with a horizontal side is considered to be a trapezoid with two horizontal sides one of which is degenerate. See figure 4. In such systems the layout is stored as a set of polygonal regions which should be partitioned into fundamental figures since the aperture of a pattern generator is restricted. Trapezoids have been used as fundamental figures. Asano et al [6] develop an $O(n^3)$ time algorithm, based on circle graphs, for the problem when the polygon does not contain holes. If a minimum partition is not important, Chazelle is able to partition a polygon into trapezoids in $O(n)$ time as a by-product of his linear time triangulation algorithm [27]. In the case where the polygon does contain holes, Asano et al [6] show the problem to be NP-complete, and they provide an $O(nlogn)$ time approximation algorithm that finds a partition containing not more than three times the number of trapezoids in a minimum partition.

Everett et al [40] consider the problem of partitioning a polygon into convex quadrilaterals. They use Steiner points, and give an $O(n)$ time algorithm that is not guaranteed to provide the minimum number of components. Another $O(n)$ time algorithm for this problem, that limits the number of Steiner points, is given in [116]. It is not always possible to partition a polygon into convex quadrilaterals without adding Steiner points. Lubiw [91] shows that the problem of deciding whether or not a partition without Steiner points is possible is NP-complete. Algorithms for partitioning convex polygons with point holes into quadrilaterals are given in [134, 18].

Levcopoulos et al [81, 78] provide some algorithms for partitioning some types of polygons into m-gons under the minimum edge length optimization criterion.

0.2.2 Polygon Covering

Much of the work done concerning covering general polygons has involved convex or star-shaped components.

Convex Subpolygons

The problem of covering a polygon with the minimum number of convex subpolygons finds application in syntactic pattern recognition [41, 110, 112, 111, 109]; for example in the recognition of chinese characters. In 1982 O'Rourke was one of the first to investigate the complexity of this problem. He showed that, although it is difficult to restrict the possible locations of Steiner points [105], the problem is nevertheless decidable [104, 103]. For polygons with holes, O'Rourke and Supowit show that the problem is NP-hard [109], with or without Steiner points, and for this problem O'Rourke[106] provides an algorithm which runs in exponential time. Several years later, in sharp contrast to the partitioning situation, Culberson and Reckhow show that even if the polygon does not contain holes, the problem of covering a polygon with the minimum number of convex components remains NP-hard [35].

The difficulty of the problem motivates the consideration of the problem of covering a polygon with a fixed number of convex subpolygons. Shermer [129] provides a linear time algorithm for recognizing polygons that can be covered with two convex subpolygons. Belleville provides a linear time algorithm for recognizing polygons that can be covered with three convex subpolygons [13, 14].

A more general type of polygon decomposition allows set difference as well as union as an operator to apply to the components. This additional operator may allow for a smaller number of component pieces. Batchelor [12] investigates a procedural approach to convex *sum/difference* decompositions. This type of decomposition has been applied to the automatic transformation of sequential programs for efficient execution on parallel computers [94]. Also, Tor and Middleditch [131] give an $O(n^2)$ time algorithm for finding a convex sum/difference decomposition that may not necessarily use the minimum number of components.

Star-shaped Subpolygons

The problem of covering a polygon with star-shaped subpolygons has often been investigated as the problem of guarding an art gallery [9, 107, 128]. The region visible from a guard is a star-shaped subpolygon and the polygon models the art gallery. Knowledge of this problem helps with the understanding of visibility problems within polygons.

General satisfactory solutions are not known for the minimum star-shaped covering problem. In 1983 for polygons with holes, O'Rourke and Supowit show the problem to be NP-hard [109]. Later Lee and Lin [70] show the problem remains NP-hard, even without holes, if the kernel of each star-shaped subpolygon must contain a vertex. Aggarwal [1] then shows that the unrestricted problem is NP-hard for polygons without holes. More recently other variations of the problem are shown to be NP-hard by Hecker and Herwig [59], and by Nilsson [101].

In 1987 Ghosh [48] develops an $O(n^5 logn)$ time approximation algorithm, that finds a cover within a factor of $O(logn)$ of optimal, if the kernels of the subpolygons are restricted to contain vertices. His algorithm works whether or not the polygon contains holes. In 1988 Aggarwal et al [3] consider a restricted problem, for polygons without holes, where subpolygon sides must be contained in either edges, edge extensions or segments of lines passing through two vertices of the polygon. For this restricted problem they develop an $O(n^4 logn)$ time approximation algorithm that produces a cover within a factor of $O(logn)$ of optimal. They also show that the restricted problem remains NP-hard.

Belleville [15] investigates the problem of recognizing polygons that can be covered by two star-shaped subpolygons. He gives an $O(n^4)$ time algorithm for recognizing such polygons.

Shermer [127] contributes to knowledge of related problems by giving bounds on the number of generalized star-shaped components required in a generalized cover.

Other Subpolygons

Spiral polygons and rectangles are two other types of component subpolygons that have been used to cover a polygon. For polygons with holes, O'Rourke and Supowit [109] show that covering with the minimum number of spiral subpolygons is NP-hard.

Levcopoulos and Lingas [77] consider covering acute polygons, whose interior angles are all greater than $90°$, by rectangles. They show that for convex polygons, the minimum number of rectangles needed in a cover is $O(nlog(r(P)))$, where $r(P)$ is the ratio of the length of the longest edge of the polygon to the length of the shortest edge of the polygon. Later Levcopoulos [71, 74] extends this result and gives an algorithm that covers such an acute polygon with $O(nlogn + m(P))$ rectangles in time $O(nlogn + m(P))$, where $m(P)$ is the number of rectangles in an optimal cover.

0.3 Orthogonal Polygons

In this section we turn our attention to the problem of decomposing orthogonal polygons. An orthogonal polygon is a polygon whose edges are either horizontal or vertical. Orthogonal polygons are also referred to as rectilinear polygons. They arise in applications, such as image processing and VLSI design, where a polygon is stored relative to an implicit grid. The set of orthogonal polygons is a subset of the set of all polygons, thus any polynomial time algorithm developed for general polygons will apply to orthogonal polygons, but problems NP-complete for general polygons may become tractable when restricted to orthogonal polygons. There are also natural subpolygons for orthogonal polygons, such as axis aligned rectangles or squares that are less relevant to general polygons. In the next two subsections we treat partitioning and covering problems for orthogonal polygons.

0.3.1 Partitioning Orthogonal Polygons

Rectangles are the most important type of component to consider in relation to orthogonal polygons. The problem of partitioning orthogonal polygons into axis aligned rectangles has many applications.

Image processing is often more efficient when the image in rectangular. For example, Ferrari et el [42] indicate that the convolving of an image with a point spread function can be made particularly efficient by specifying the nonnegative values of the point spread function over a rectangular domain and specifying that function to be zero outside that domain. They suggest handling a nonrectangular orthogonal image by partitioning it into the minimum number of rectangular subregions.

In VLSI design, two variations of the problem arise. The first occurs in optimal automated VLSI mask fabrication [83, 100, 102]. In mask generation a figure is usually engraved on a piece of glass using a pattern generator. A traditional pattern generator has a rectangular opening, thus the figure must be partitioned into rectangles so that the pattern generator can expose each such rectangle. The entire figure can be viewed as an orthogonal polygon. Since the time required for mask generation depends on the number of rectangles, the problem of partitioning an orthogonal polygon into the minimum number of rectangles becomes relevant. Another VLSI design problem is that of dividing the routing region into channels. Lingas et al [82] suggest that partitioning the orthogonal routing region into rectangles, while minimizing the total length of the lines used to form the decomposition, will produce large "natural-looking" channels with a minimum of channel to channel interaction. Thus the minimum "ink" criteria is also relevant.

Other application areas for the problem of partitioning orthogonal polygons into the minimum number of rectangles include database systems [90] and computer graphics [54].

At this point we should note that the use of Steiner points is inherent in the solution of the problem of partitioning into rectangles. For example for an orthogonal polygon with one reflex vertex, a partition can be formed by adding a horizontal line segment from the reflex vertex to the polygon boundary. In fact a generalization of this idea forms the basis for most partitioning algorithms. The following theorem [87, 102, 42] expresses this. See figure 5.

Theorem 1 *An orthogonal polygon can be minimally partitioned into $N - L - H + 1$ rectangles, where N is the number of reflex vertices, H is the number of holes and L is the maximum number of nonintersecting chords that can be drawn either horizontally or vertically between reflex vertices.*

The theorem implies that a key step in the decomposition problem is that of finding the maximum number of independent vertices in the intersection graph of the vertical or horizontal chords between reflex vertices. This problem can in turn be solved by finding a maximum matching in a bipartite graph. In 1979 Lipski et al [87] exploited this approach to develop an $O(n^{\frac{5}{2}})$ time algorithm for partitioning orthogonal polygons with holes. Algorithms for the same problem running within the same time bounds were also developed in [102] and [42]. In the early 1980s the special structure of the bipartite graph involved allowed the development of improved algorithms for the problem running in $O(n^{\frac{3}{2}} logn)$ time [62, 85, 86]. These algorithms have been recently extended by Soltan and Gorpinevich [130] to run in the same time bounds even if the holes degenerate to points. It is open as to whether or not faster algorithms can be developed. The only known lower bound for the problem with holes is $\Omega(nlogn)$ [83].

If the polygons do not contain holes then faster algorithms are possible. In 1983 Gourley and Green [54] develop an $O(n^2)$ time algorithm that partitions an orthogonal polygon without holes into within 3/2 of the minimum number of rectangles. In 1988 Naha and Sahni [100] also develop an algorithm that partitions into less than 3/2 of the minimum number of rectangles, but their algorithm runs in $O(nlogn)$ time. Finally in 1989, Liou et al [83] produce an $O(n)$ time algorithm to optimally partition an orthogonal polygon without holes into the minimum number of rectangles. The $O(n)$ time is achieved assuming that the polygon is first triangulated using Chazelle's linear time triangulation algorithm.

Note that the three dimensional version of the problem is NP-complete [36].

Minimizing the total length of the line segments introduced in the partitioning process is the other optimization criterion that arises in the applications. See figure 6. Lingas et al [82] were the first to investigate this optimization criteria. They present an $O(n^4)$ time algorithm for the problem of partitioning an orthogonal polygon without holes into rectangles using the minimum amount of ink. If the polygon contains holes they show that the problem becomes NP-complete.

In applications holes do occur, thus the search was on for approximation algorithms for the problem. The first algorithm of this type was given by Lingas [80] . In 1983 he presented an $O(n^4)$ time algorithm to partition an orthogonal polygon with holes into rectangles such that the amount of "ink" used is within a constant of the minimum amount possible. Unfortunately, the constant for this algorithm is large (41). In 1986 Levcopoulos [73] was able to reduce the constant to five while also producing an algorithm running in only $O(n^2)$ time. In the same year [72] he further reduced the time to $O(nlogn)$, but at the expense of a large increase in the size of the constant.

The restriction of the problem to where the orthogonal polygon becomes a rectangle and the holes become points is also NP-complete [82]. Gonzalez and Zheng [51] show how to adapt any approximation algorithm for the restricted problem to yield an approximation algorithm for the more general problem where the boundary polygon need not be a rectangle. Their method is to use the algorithm given in [82] to partition the boundary orthogonal polygon into rectangles, then each of these rectangles along with the point holes inside it, becomes an instance of the restricted version of the problem.

In 1985 Gonzalez and Zheng [51] give an approximation algorithm, running in $O(n^2)$ time that partitions a rectangle with point holes into disjoint rectangles, using no more than $3+\sqrt{3}$ times the minimum amount of "ink" required. The next year Levcopoulos [72] improves the time to $O(nlogn)$, while maintaining the same bound. Later Gonzalez and Zheng [53] give an algorithm that runs in $O(n^4)$ time, that produces a solution within 3 times optimal. They use a so called "guillotine" partition to develop an approximation algorithm within 1.75 times optimal [52], but which uses $O(n^5)$ time. See figure 7. A recent paper [49] provides a simpler proof that the "guillotine" partition is within 2 times optimal. If time is more important, Gonzalez et al [50] give an algorithm that runs in $O(nlogn)$ time, but only finds a solution guaranteed to be within four times optimal.

If Steiner points are disallowed, then quadrilaterals rather than rectangles become the natural component type for the decomposition of orthogonal polygons. Kahn, Klawe and Kleitman [63] show that it is always possible to partition an orthogonal polygon into convex quadrilaterals. This is not always possible for arbitrary polygons. This partitioning of a polygon into convex quadrilaterals is referred to as *quadrilateralization.* Sack and Toussaint develop an $O(nlogn)$ time algorithm for quadrilateralizing an orthogonal polygon [119, 121]. They use a two step process. First the orthogonal polygon is partitioned into a specific type of monotone polygon, these are in turn partitioned into quadrilaterals in linear time [120]. Lubiw [91] also provides an $O(nlogn)$ time quadrilateralization algorithm for orthogonal polygons. Arbitrary monotone or star-shaped orthogonal polygons can be quadrilateralized in linear time [121].

Let us now turn to the problem of finding the minimum edge length quadrilateralization of an orthogonal polygon. For this problem Keil and Sack [68] give an $O(n^4)$ time algorithm. Later Conn and O'Rourke [32] improve this to $O(n^3 logn)$ time.

There are other known results concerning orthogonal partitioning. Liu and Ntafos [89] show how to partition a monotone orthogonal polygon into star-shaped components. Their algorithm runs in $O(nlogn)$ time, allows Steiner points, and yields a decomposition within four times optimal. Gunther [56] gives a polynomial time algorithm for partitioning an orthogonal polygon into orthogonal polygons with k or fewer vertices. In most cases this algorithm finds a partition that is within a factor of two of optimal. Györi et al [58] also have some results on partitioning orthogonal polygons into subpolygons with a fixed number of vertices.

0.3.2 Covering Orthogonal Polygons

Tools from graph theory are useful when developing algorithms for covering orthogonal polygons. If each edge of an orthogonal polygon is extended to a line, a rectangular grid results. Based on this grid, a graph can be associated with a covering problem as follows. The vertices of the graph are the grid squares that lie within the polygon and two such vertices are adjacent if the associated grid squares can be covered by a subpolygon lying entirely within the polygon. Depending upon the type of subpolygon, there can be a correspondence between covering the graph with the minimum number of cliques and the original polygon covering problem. For example in figure 8, if the grid squares are joined by edges if they lie in the same rectangle, then the problem of covering the polygon with rectangles corresponds to covering the derived graph with cliques. If such a correspondence exists, then the tractability of both problems depends upon the properties of the derived graph. This graph theory approach underlies several of the algorithms we shall encounter in this section. The types of subpolygons that have been studied include rectangles, squares, orthogonally convex, orthogonally star-shaped and others.

Rectangles

The problem of covering an orthogonal polygon with the minimum number of axis aligned rectangles has found application in data compression [90], the storing of graphic images [95], and the manufacture of integrated circuits [23, 60].

As early as 1979 Masek [95] showed that if the orthogonal polygon contains holes, then the problem is NP-complete. Later Conn and O'Rourke [31] show that for an orthogonal polygon with holes it is also NP-complete if only the boundary or only the reflex vertices need to be covered. Attention then turned to the case where the polygon does not contain holes.

In 1981 Chaiken et al [23] initiated the above mentioned graph theory approach. They define a graph G with grid squares for vertices and with two vertices adjacent if there is a rectangle, lying entirely within the polygon, that contains both associated grid squares. They show that the cliques of this graph correspond to the rectangles in the polygon whose sides lie on grid lines. The rectangle cover problem then corresponds to the problem of covering the vertices of the graph with the minimum number of cliques. This clique problem is NP-complete in general but polynomially solvable for the class of perfect graphs. Unfortunately, the graph derived from the rectangle problem is not perfect, even if the polygon does not contain holes [23].

In the search for a solvable restriction of the problem attention turned to restricted types of orthogonal polygons. An orthogonal polygon is called *horizontally (vertically) convex* if its intersection with every horizontal (vertical) line is either empty or a single line segment. For an example see figure 9. An orthogonal polygon is called *orthogonally convex* if it is both horizontally and vertically convex. Chaiken et al [23] have an example showing that even for orthogonally convex polygons the derived graph is not perfect. Thus for the rectangle covering problem the graph approach has not yielded efficient algorithms. Note that the intersection graph of the maximal rectangles in an orthogonal polygon without holes is perfect [126].

To develop a polynomial time algorithm for the special case of covering an orthogonally convex polygon with the minimum number of rectangles, in 1981 Chaiken et al [23] use an approach that reduces the problem to the same problem on a smaller polygon. Later Liou et al [84] develop an $O(n)$ time algorithm for this problem. Brandstadt also contributes a linear algorithm for the restricted case of 2-staircase polygons [19].

In 1984 Franzblau and Kleitman [43] handle the larger class of horizontally convex polygons. They give a polynomial time algorithm for covering this class with the minimum number of rectangles. See also [57].

In 1985 Lubiw [92] was able to provide a polynomial time algorithm for another restricted class of orthogonal polygons. Her algorithm handles orthogonal polygons that do not contain a rectangle that touches the boundary of the polygon only at two opposite corners of the rectangle.

In spite of these efforts on special cases the general problem of covering an orthogonal polygon without holes with the minimum number of rectangles remained open for some time. Finally in 1988, Culberson and Reckhow [35] settle the issue by showing the problem to be NP-complete. Later Berman and Das Gupta [16] go further and show that no polynomial time approximation scheme for the problem exists unless P = NP.

The difficulty of the problem led Cheng et al [30], in 1984, to develop a linear approximation algorithm that is guaranteed to find a solution within four times optimal for hole free polygons. Then in 1989 Franzblau develops an $O(nlogn)$ time approximation algorithm that yields a covering containing $O(\theta log\theta)$ rectangles, where θ is the minimum number of rectangles required for a covering [44]. She also shows that an optimal partitioning contains at most $2\theta + H - 1$ rectangles, where H is the number of holes contained in the polygon.

Recently, Keil [67] introduces a type of rectangle decomposition which is intermediate between partitioning and covering. This non-piercing covering allows rectangles to overlap, but if two rectangles A and B overlap, then either $A - B$ or $B - A$ must be connected. Keil provides an $O(nlogn + mn)$ time algorithm for finding an optimal non-piercing covering of an orthogonal polygon P without holes, where m is the number of edges in the visibility graph of P that are either horizontal, vertical or form the diagonal of an empty rectangle.

Squares

Covering polygons with axis aligned squares has application in the construction of data structures used in the storage and processing of digital images. For example, the digital medial axis transform (MAT) [135] is based on representing an image by the union of squares. Simple images may be covered by few squares and may be easily reconstructed from the MAT.

Scott and Iyengar [124] define the Translation Invariant Data Structure (TID), as a method for representing images. An image is considered to be a grid of "black" and "white" pixels, and the TID for a given image consists of a list of maximal squares covering all black regions within the image. In order to reduce the cost of storing and manipulating a TID, the underlying list of squares should be as small as possible. Scott and Iyengar [124] give a heuristic for finding a small covering set of squares as part of their TID construction algorithm.

Albertson and O'Keefe [5] investigate a graph associated with the square covering problem. A unit square in the plane whose corners are integer lattice points is called a block. A polygon with integer vertices then contains of set of N blocks. Albertson and O'Keefe define a graph, with the blocks as vertices and with such vertices adjacent if the corresponding blocks can be covered by a square lying entirely within the polygon. They show that for polygons without holes this graph is perfect. They further show that the blocks corresponding to a clique in the graph form a set of blocks entirely contained within a single square lying in the polygon. Aupperle et al [8] investigate this graph further and show that for polygons without holes the graph is chordal. This allows the use of an algorithm for covering chordal graphs by cliques to serve as an $O(N^{2.5})$ time algorithm for the problem of covering an orthogonal polygon without holes with the minimum number of squares. By further exploiting the geometry, Aupperle [7] adapts this approach to produce an $O(N^{1.5})$ time algorithm for this problem. The fastest algorithm based on the blocks lying in the polygon runs in $O(N)$ time and is due to Moitra [96, 97].

The number N of blocks lying in the polygon could be much larger that n, the number of vertices defining the polygon. Even if the block side is optimized N may be $\Omega(n^2)$. In light of this Bar-Yeuda and Ben-Chanoch [11] consider an alternative approach of covering the polygon one square at a time and achieve an $O(n + \theta)$ time algorithm for covering an orthogonal polygon without holes, where θ is the minimum number of squares required in a cover.

If the polygon contains holes the square coverage problem becomes NP-complete [8, 7].

Orthogonal Convex and Star-shaped subpolygons

When restricting the polygons to be orthogonal we find it is natural to also restrict the notion of visibility. We consider two notions of orthogonal visibility [33]. Two points of a polygon are said to be r-visible if there exists a rectangle that contains the two points [66]. Using r-visibility leads to the fact that an r-convex polygon is just a rectangle. The decomposition of an orthogonal polygon into rectangles has been discussed in previous subsections. An r-star-shaped polygon, P, is an orthogonal polygon for which there exists a point q of P such that for all other points p of P p is r-visible (ie. lies in the same rectangle) to q.

Recall that an orthogonally convex polygon is an orthogonal polygon that is both horizontally and vertically convex. Two points of an orthogonal polygon are said to be s-visible (staircase visible) if there exists an orthogonally convex subpolygon containing both points. Note that an s-convex polygon is simply an orthogonally convex polygon. An s-starshaped polygon contains a point q, such that for every point p, in the polygon, there is an orthogonally convex subpolygon containing both p and q.

In this subsection we will consider the problems of covering an orthogonal polygon with the minimum number of r-stars, s-stars and orthogonally convex polygons.

A classification of orthogonal polygons, due to Reckhow and Culberson [118], based on the types of "dents" encountered, has been useful in the work on these problems [34, 33, 99, 98, 115, 117, 118]. If the boundary of the orthogonal polygon is traversed in the clockwise direction, at each corner either a right 90° (outside corner) or a left 90° (inside corner) turn is made. A *dent* is an edge of P both of whose endpoints are inside corners. If the polygon is aligned so that north (N) corresponds with the positive y axis, then dents can be classified according to compass directions. For example, an N dent is traversed from west to east in a clockwise traversal of the polygon. An orthogonal polygon can then be classified according to the number and the types of dents it contains. A class k orthogonal polygon contains dents of k different orientations. Class 0 orthogonal polygons are the orthogonally convex polygons. A vertically or horizontally convex polygon (class 2a) is a class 2 orthogonal polygon which has only opposing pairs of dent types (ie N and S or E and W). Class 2b orthogonal polygons have two dent orientations that are orthogonal to one another (ie. W and N, N and E, E and S, or S and W). For an example of a class 3 polygon see figure 10.

The graph theory approach has been important in the development of understanding of these problems. By extending the dent edges across the polygon a partition into $O(n^2)$ basic regions results. These basic regions correspond to vertices in the definition of several relevant graphs. Motwani et al [99, 98] define an s-convex visibility graph, using the basic regions as vertices, where two vertices are adjacent in the graph if the corresponding basic regions can be covered by a single orthogonally convex subpolygon. They define an r-star (s-star) visibility graph, again using the basic regions as vertices, where two vertices u and v are adjacent if there is a region w that is r-visible (s-visible) to the regions corresponding to u and v. Related directed graphs were defined by Culberson and Reckhow [118, 33, 117, 34].

In 1986 Keil [66] provides the first algorithm for minimally covering with orthogonally convex components. He provides an $O(n^2)$ time algorithm for covering horizontally convex orthogonal polygons. In 1987, for this problem, Reckhow and Culberson [118, 34] give an $\Omega(n^2)$ lower bound on actually listing the vertices of all the subpolygons in the output, but provide an $O(n)$ time algorithm for finding the minimum number of orthogonally convex polygons in an optimal cover of a horizontally convex orthogonal polygon. Culberson and Reckhow also provide an $O(n^2)$ algorithm for minimally covering class 2b type orthogonal polygons, and they give a complex algorithm for handling a larger class. Later for class 3 polygons Reckhow [117] provides an $O(n^2)$ time algorithm.

For the problem of covering with orthogonally convex components the relevant convex visibility graph is formed by connecting two grid squares if they can be covered by a single orthogonally convex subpolygon [98, 115, 117, 118]. Motwani et al [98] prove that a minimum clique cover of this visibility graph corresponds exactly to a minimum cover of the corresponding orthogonal polygon by orthogonal convex polygons. Thus we may solve the polygon covering problem using existing graph clique cover algorithms. The complexity of the available clique cover algorithms depends upon the properties of the convex visibility graph. For class 2 polygons, the convex visibility graph is a permutation graph [98, 115]. For class 3 polygons, the graph turns out to be weakly triangulated [98, 117]. Although these graph classes allow polynomial time clique cover algorithms, the known geometric algorithms are still the most efficient algorithms to solve the problem of covering an orthogonal polygon with the minimum number of orthogonally convex subpolygons. For class 4 polygons (general orthogonal polygons) the convex visibility graph is not perfect [98, 115] and the complexity of the general problem of covering orthogonal polygons with the minimum number of orthogonally convex subpolygons remains open.

For covering with r-stars, Keil [66] provides an $O(n^2)$ time algorithm for optimally covering a horizontally convex orthogonal polygon. This is later improved to $O(n)$ time in [46]. For general class 2 polygons, class 3 polygons and general orthogonal polygons the problem of covering with the minimum number of r-stars remains open.

For covering with s-stars, Culberson and Reckhow [33] provide $O(n^2)$ time algorithms for optimally covering horizontally convex orthogonal polygons and general class 2 polygons. For class 3 and class 4 polygons the development of algorithms has depended upon properties of the s-star visibility graph. Motwani et al [99] show that for class 3 polygons the derived s-star visibility graph is chordal. They show that a minimum clique cover algorithm for chordal graphs can be used to s-star cover class 3 polygons in $O(n^3)$ time. For class 4 polygons they show that the derived s-star graph is weakly triangulated. This then leads to an $O(n^8)$ time algorithm for the general s-star covering problem [99].

Other covering subpolygons

There are some other known results related to the covering of orthogonal polygons with subpolygons. Bremner and Shermer [20, 21] studied an extension of orthogonal visibility, called $\mathcal{O}$-visibility, in which two points of the polygon are $\mathcal{O}$-visible if there is a path between them whose intersection with every line in the set $\mathcal{O}$ of orientations is either empty of connected. For orthogonal visibility $\mathcal{O} = \{0°, 90°\}$. A polygon P is $\mathcal{O}$-convex if every two points of P are $\mathcal{O}$-visible, and $\mathcal{O}$-star-shaped if there is a point of P from which every other point of P is $\mathcal{O}$-visible. Bremner and Shermer [20, 21] were able to characterize classes of orientations for which minimum covers of a (not necessarily orthogonal) polygon by $\mathcal{O}$-convex or $\mathcal{O}$-star-shaped components could be found in polynomial time.

Regular star-shaped (nonorthogonal) polygons have also been studied as covering subpolygons. Edelsbrunner et al [38] give an $O(nlogn)$ time algorithm that covers an orthogonal polygon with $\lfloor \frac{r}{2} \rfloor + 1$ star-shaped subpolygons, where r is the number of reflex vertices in the polygon. This is improved by Sack and Toussaint [122] who give an $O(n)$ time algorithm for covering an orthogonal polygon with $\lfloor \frac{n}{4} \rfloor$ star-shaped components. Carlsson et al [22] are able to produce an optimal star-shaped cover for histograms in linear time.

Gewali and Ntafos [47] consider covering with a variation of r-stars where periscope vision is allowed. They give an $O(n^3)$ algorithm for optimally covering a restricted type of orthogonal polygon. A different variation of r-stars is considered by Maire [93]. His stars consist of a union of a vertical and a horizontal rectangle and look like "plus" signs. He defines a type of star graph and shows it is weakly triangulated implying an optimal algorithm for the covering problem.

Acknowledgement

The author wishes to thank the Natural Sciences and Engineering Research Council of Canada for financial support.

Bibliography

[1] A. Aggarwal. *The art gallery problem: Its variations, applications, and algorithmic aspects.* Ph.D. thesis, Johns Hopkins Univ., Baltimore, MD, 1984.

[2] A. Aggarwal and B. Chazelle. Efficient algorithm for partitioning a polygon into star-shaped polygons. Report, IBM T. J. Watson Res. Center, Yorktown Heights, NY, 1984.

[3] A. Aggarwal, S. K. Ghosh, and R. K. Shyamasundar. Computational complexity of restricted polygon decompositions. In G. T. Toussaint, editor, *Computational Morphology*, pages 1–11. North-Holland, Amsterdam, Netherlands, 1988.

[4] A. Aggarwal, L. J. Guibas, J. Saxe, and P. W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete Comput. Geom.*, 4(6):591–604, 1989.

[5] M. Albertson and C. J. O'Keefe. Covering regions by squares. *SIAM J. Algebraic Discrete Methods*, 2(3):240–243, 1981.

[6] Ta. Asano, Te. Asano, and H. Imai. Partitioning a polygonal region into trapezoids. *J. ACM*, 33:290–312, 1986.

[7] L. J. Aupperle. Covering regions by squares. M.Sc. thesis, Dept. Comput. Sci., Univ. Saskatchewan, Saskatoon, Sask., 1987.

[8] L. J. Aupperle, H. E. Conn, J. M. Keil, and J. O'Rourke. Covering orthogonal polygons with squares. In *Proc. 26th Allerton Conf. Commun. Control Comput.*, pages 97–106, October 1988.

[9] D. Avis and G. T. Toussaint. An efficient algorithm for decomposing a polygon into star-shaped polygons. *Pattern Recogn.*, 13:395–398, 1981.

[10] C. Bajaj and T. K. Dey. Convex decompositions of polyhedra and robustness. *SIAM J. on Computing*, 21:339–364, 1992.

[11] R. Bar-Yeuda and E. Ben-Chanoch. A linear time algorithm for covering simple polygons with similar rectangles. *Int. J. of Computational Geometry and Applications*, 6:79–102, 1996.

[12] B. G. Batchelor. Hierarchical shape description based upon convex hulls of concavities. *J. Cybern.*, 10:205–210, 1980.

[13] P. Belleville. On restricted boundary covers and convex three-covers. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 467–472, Waterloo, Canada, 1993.

[14] P. Belleville. *A Study of Convex Covers in Two or More Dimensions*. Ph.D. thesis, School of Computing Science, Simon Fraser University, 1995.

[15] Patrice Belleville. Computing two-covers of simple polygons. Master's thesis, School of Computer Science, McGill University, 1991.

[16] P. Berman and B. DasGupta. Approximating the rectilinear polygon cover problems. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 229–235, 1992.

[17] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 23–90. World Scientific, Singapore, 1992.

[18] P. Bose and G. Toussaint. Generating quadrangulations of planar point sets. *Computer Aided Geometric Design*, 1996.

[19] A. Brandstadt. The jump number problem for biconvex graphs and rectangle covers of rectangular regions. *Lecture Notes in Computer Science*, 380:68–77, 1989.

[20] David Bremner. Point visibility graphs and restricted-orientation polygon covering. M.Sc. Thesis, School of Computing Science, Simon Fraser University, Burnaby, BC, April 1993.

[21] David Bremner and Thomas Shermer. Point visibility graphs and restricted-oriention convex cover. Technical Report CMPT TR 93-07, School of Computing Science, Simon Fraser University, 1993.

[22] S. Carlsson, B. J. Nilsson, and S. Ntafos. Optimum guard covers and m-watchmen routes for restricted polygons. In *Proc. 2nd Workshop Algorithms Data Struct.*, volume 519 of *Lecture Notes in Computer Science*, pages 367–378. Springer-Verlag, 1991.

[23] S. Chaiken, D. Kleitman, M. Saks, and J. Shearer. Covering regions by rectangles. *SIAM J. Algebraic Discrete Methods*, 2:394–410, 1981.

[24] B. Chazelle. *Computational geometry and convexity.* Ph.D. thesis, Dept. Comput. Sci., Yale Univ., New Haven, CT, 1979. Carnegie-Mellon Univ. Report CS-80-150.

[25] B. Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 339–349, 1982.

[26] B. Chazelle. Approximation and decomposition of shapes. In J. T. Schwartz and C.-K. Yap, editors, *Advances in Robotics 1: Algorithmic and Geometric Aspects of Robotics*, pages 145–186. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

[27] B. Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6:485–524, 1991.

[28] B. Chazelle and D. P. Dobkin. Decomposing a polygon into its convex parts. In *Proc. 11th Annu. ACM Sympos. Theory Comput.*, pages 38–48, 1979.

[29] B. Chazelle and D. P. Dobkin. Optimal convex decompositions. In G. T. Toussaint, editor, *Computational Geometry*, pages 63–133. North-Holland, Amsterdam, Netherlands, 1985.

[30] Y. Cheng, S. Iyengar, and R. Kashyap. A new method of image compression using irreducible covers of maximal rectangles. *IEEE Trans. on Software Engineering*, 14:651–658, 1988.

[31] H. Conn and J. O'Rourke. Some restricted rectangle covering problems. Technical Report JHU 87-13, Dept. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1987.

[32] H. E. Conn and J. O'Rourke. Minimum weight quadrilaterization in $O(n^3 \log n)$ time. In *Proc. 28th Allerton Conf. Commun. Control Comput.*, pages 788–797, October 1990.

[33] J. Culberson and R. A. Reckhow. Dent diagrams: A unified approach to polygon covering problems. Technical Report TR 87-14, Dept. of Computing Science, University of Alberta, 1987.

[34] J. Culberson and R. A. Reckhow. Orthogonally convex coverings of orthogonal polygons without holes. *J. Comput. Syst. Sci.*, 39:166–204, 1989.

[35] J. Culberson and R. A. Reckhow. Covering polygons is hard. *Journal of Algorithms*, 17:2–44, 1994.

[36] V. J. Dielissen and A. Kaldewaij. Rectangular partition is polynomial in two dimensions but NP-complete in three. *Inform. Process. Lett.*, 38:1–6, 1991.

[37] D. P. Dobkin, D. L. Souvaine, and C. J. Van Wyk. Decomposition and intersection of simple splinegons. *Algorithmica*, 3:473–486, 1988.

[38] H. Edelsbrunner, J. O'Rourke, and E. Welzl. Stationing guards in rectilinear art galleries. *Comput. Vision Graph. Image Process.*, 28:167–176, 1984.

[39] H. ElGindy and G. T. Toussaint. On geodesic properties of polygons relevant to linear time triangulation. *Visual Comput.*, 5(1–2):68–74, 1989.

[40] H. Everett, W. Lenhart, M. Overmars, T. Shermer, and J. Urrutia. Strictly convex quadrilateralizations of polygons. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 77–83, 1992.

[41] H. Y. F. Feng and T. Pavlidis. Decomposition of polygons into simpler components: feature generation for syntactic pattern recognition. *IEEE Trans. Comput.*, C-24:636–650, 1975.

[42] L. Ferrari, P. V. Sankar, and J. Sklansky. Minimal rectangular partitions of digitized blobs. *Computer Vision, Graphics, and Image Processing*, 28:58–71, 1984.

[43] D. Franzblau and D. Kleitman. An algorithm for covering polygons with rectangles. *Inform. Control*, 63:164–189, 1984.

[44] D. S. Franzblau. Performance guarantees on a sweep-line heuristic for covering rectilinear polygons with rectangles. *SIAM Journal on Discrete Mathematics*, 2:307–321, 1989.

[45] M. R. Garey, D. S. Johnson, F. P. Preparata, and R. E. Tarjan. Triangulating a simple polygon. *Inform. Process. Lett.*, 7:175–179, 1978.

[46] L. Gewali, J. M. Keil, and S. Ntafos. On covering orthogonal polygons with star shaped polygons. *Information Sciences*, 65:45–63, 1992.

[47] L. Gewali and S. Ntafos. Minimum covers for grids and orthogonal polygons by periscope guards. In *Proc. 2nd Canad. Conf. Comput. Geom.*, pages 358–361, 1990.

[48] S. K. Ghosh. Approximation algorithms for art gallery problems. In *Proc. Canadian Inform. Process. Soc. Congress*, 1987.

[49] T. Gonzalez, M. Razzazi, M.-T. Shing, and S.-Q. Zheng. On optimal guillotine partitions approximating optimal d-box partitions. *Computational Geometry: Theory and Applications*, 4:1–12, 1994.

[50] T. Gonzalez, M. Razzazi, and S.-Q. Zheng. An efficient divide-and-conquer approximation algorithm for partitioning into d-boxes. *Inter. Journal of Computational Geometry and Applications*, 3:417–428, 1993.

[51] T. Gonzalez and S.-Q. Zheng. Bounds for partitioning rectilinear polygons. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 281–287, 1985.

[52] T. Gonzalez and S.-Q. Zheng. Improved bounds for rectangular and guillotine partitions. *J. Symbolic Comput.*, 7:591–610, 1989.

[53] T. Gonzalez and S.-Q. Zheng. Approximation algorithms for partitioning a rectangle with interior points. *Algorithmica*, 5:11–42, 1990.

[54] K. D. Gourley and D. M. Green. A polygon-to-rectangle conversion algorithm. *IEEE Computer Graphics*, 3:31–36, 1983.

[55] D. H. Greene. The decomposition of polygons into convex parts. In F. P. Preparata, editor, *Computational Geometry*, volume 1 of *Advances in Computing Research*, pages 235–259. JAI Press, London, England, 1983.

[56] O. Gunther. Minimum k-partitioning of rectilinear polygons. *J. of Symbolic Computation*, 9:457–483, 1990.

[57] E. Gyori. Covering simply connected regions by rectangles. *Combinatorica*, 5:53–55, 1985.

[58] E. Györi, F. Hoffmann, K. Kriegel, and T. Shermer. Generalized guarding and partitioning for rectilinear polygons. In *Proc. 6th Canad. Conf. Comput. Geom.*, pages 302–307, 1994.

[59] H.-D. Hecker and D. Herwig. Some NP-hard polygon cover problems. *J. Inf. Process. Cybern.*, 25:101–108, 1989.

[60] A. Hegedus. Algorithms for covering polygons by rectangles. *Computer Aided Geometric Design*, 14:257–260, 1982.

[61] S. Hertel and K. Mehlhorn. Fast triangulation of the plane with respect to simple polygons. *Inform. Control*, 64:52–76, 1985.

[62] H. Imai and Ta. Asano. Efficient algorithms for geometric graph search problems. *SIAM J. Comput.*, 15:478–494, 1986.

[63] J. Kahn, M. Klawe, and D. Kleitman. Traditional galleries require fewer watchmen. *SIAM J. Algebraic Discrete Methods*, 4:194–206, 1983.

[64] J. M. Keil. *Decomposing a polygon into simpler components.* Ph.D. thesis, Univ. of Toronto, Toronto, Canada, 1983. Report 163/83.

[65] J. M. Keil. Decomposing a polygon into simpler components. *SIAM J. Comput.*, 14:799–817, 1985.

[66] J. M. Keil. Minimally covering a horizontally convex orthogonal polygon. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 43–51, 1986.

[67] J. M. Keil. Covering orthogonal polygons with non-piercing rectangles. *Inter. J. of Computational Geometry and Applications*, 1996.

[68] J. M. Keil and J.-R. Sack. Minimum decompositions of polygonal objects. In G. T. Toussaint, editor, *Computational Geometry*, pages 197–216. North-Holland, Amsterdam, Netherlands, 1985.

[69] G. T. Klincsek. Minimal triangulations of polygonal domains. *Discrete Math.*, 9:121–123, 1980.

[70] D. T. Lee and A. Lin. Computational complexity of art gallery problems. *IEEE Trans. Inform. Theory*, 32:276–282, 1986.

[71] C. Levcopoulos. A fast heuristic for covering polygons by rectangles. In *Proc. Fundamentals of Comput. Theory*, volume 199 of *Lecture Notes in Computer Science*. Springer-Verlag, 1985.

[72] C. Levcopoulos. Fast heuristics for minimum length rectangular partitions of polygons. In *Proc. 2nd Annu. ACM Sympos. Comput. Geom.*, pages 100–108, 1986.

[73] C. Levcopoulos. Minimum length and thickest-first rectangular partitions of polygons. Report LITH-IDA-R-86-01, Dept. of Computer and Information Science, Linkoping University, Linkoping, Sweden, 1986.

[74] C. Levcopoulos. Improved bounds for covering general polygons with rectangles. In *Proc. Conf. Found. Softw. Tech. Theoret. Comput. Sci.*, volume 287 of *Lecture Notes in Computer Science*, pages 95–102. Springer-Verlag, 1987.

[75] C. Levcopoulos and D. Krznaric. Quasi-greedy triangulations approximating the minimum weight triangulation. In *Proc. 7nd Annu. ACM-SIAM Symp. on Discrete Algorithms*, pages 392–401, 1996.

[76] C. Levcopoulos and A. Lingas. Bounds on the length of convex partitions of polygons. In *Proc. 4th Conf. Found. Softw. Tech. Theoret. Comput. Sci.*, volume 181 of *Lecture Notes in Computer Science*, pages 279–295. Springer-Verlag, 1984.

[77] C. Levcopoulos and A. Lingas. Covering polygons with minimum number of rectangles. In *Proc. 1st Sympos. Theoret. Aspects Comput. Sci.*, volume 166 of *Lecture Notes in Computer Science*, pages 63–72. Springer-Verlag, 1984.

[78] C. Levcopoulos, A. Lingas, and J.-R. Sack. Heuristics for optimum binary search trees and minimum weight t riangulation problems. *Theoret. Comput. Sci.*, 66(2):181–203, August 1989.

[79] A. Lingas. The power of non-rectilinear holes. In *Proc. 9th Internat. Colloq. Automata Lang. Program.*, volume 140 of *Lecture Notes in Computer Science*, pages 369–383. Springer-Verlag, 1982.

[80] A. Lingas. Heuristics for minimum edge length rectangular partitions of rectilinear figures. In *Proc. 6th GI Conf. Theoret. Comput. Sci.*, volume 145 of *Lecture Notes in Computer Science*, pages 199–210. Springer-Verlag, 1983.

[81] A. Lingas, C. Levcopoulos, and J.-R. Sack. Algorithms for minimum length partitions of polygons. *BIT*, 27:474–479, 1987.

[82] A. Lingas, R. Pinter, R. Rivest, and A. Shamir. Minimum edge length partitioning of rectilinear polygons. In *Proc. 20th Allerton Conf. Commun. Control Comput.*, pages 53–63, 1982.

[83] W. T. Liou, J.J.M. Tan, and R.C.T. Lee. Minimum partitioning simple rectilinear polygons in $O(nloglogn)$ time. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 344–353, 1989.

[84] W. T. Liou, C. Y. Tang, and R. C. T. Lee. Covering convex rectilinear polygons in linear time. *Internat. J. Comput. Geom. Appl.*, 1(2):137–185, 1991.

[85] W. Lipski, Jr. Finding a manhattan path and related problems. *Networks*, 13:399–409, 1983.

[86] W. Lipski, Jr. An $O(n \log n)$ Manhattan path algorithm. *Inform. Process. Lett.*, 19:99–102, 1984.

[87] W. Lipski, Jr., E. Lodi, F. Luccio, C. Mugnai, and L. Pagli. On two-dimensional data organization II. *Fundam. Inform.*, 2:245–260, 1979.

[88] R. Liu and S. Ntafos. On decomposing polygons into uniformly monotone parts. *Information Processing Letters*, 27:85–89, 1988.

[89] R. Liu and S. Ntafos. On partitioning rectilinear polygons into star-shaped polygons. *Algorithmica*, 6:771–800, 1991.

[90] E. Lodi, F. Luccio, C. Mugnai, and L. Pagli. On two-dimensional data organization I. *Fundam. Inform.*, 2:211–226, 1979.

[91] A. Lubiw. Decomposing polygonal regions into convex quadrilaterals. In *Proc. 1st Annu. ACM Sympos. Comput. Geom.*, pages 97–106, 1985.

[92] A. Lubiw. The boolean basis problem and how to cover some polygons by rectangles. *SIAM J. on Discrete Mathematics*, 3:98–115, 1990.

[93] F. Maire. Polyominos and perfect graphs. *Information Processing Letters*, 50:57–61, 1994.

[94] M. Manjuathaiah and D. Nicole. Accuractely representing the union of convex sections. Manuscript, 1995.

[95] W. J. Masek. Some NP-complete set covering problems. Manuscript, 1979.

[96] D. Moitra. *Efficient parallel algorithms for covering binary images*. Ph.D. thesis, Dept. Comput. Sci., Cornell Univ., Ithaca, NY, 1989. Technical Report TR-89-1013.

[97] D. Moitra. Finding a minimal cover for binary images: an optimal parallel algorithm. *Algorithmica*, 6:624–657, 1991.

[98] R. Motwani, A. Raghunathan, and H. Saran. Perfect graphs and orthogonally convex covers. *SIAM J. Discrete Math.*, 2:371–392, 1989.

[99] R. Motwani, A. Raghunathan, and H. Saran. Covering orthogonal polygons with star polygons: the perfect graph approach. *J. Comput. Syst. Sci.*, 40:19–48, 1990.

[100] S. Nahar and S. Sahni. Fast algorithm for polygon decomposition. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7:473–483, 1988.

[101] B. Nilsson. *Guarding Art Galleries*. Ph.D. thesis, Dept. Comput. Sci., Lund Univ., Lund, Sweden, 1994.

[102] T. Ohtsuki. Minimum dissection of rectilinear regions. In *Proceedings of the 1982 IEEE International Symposium on Circuits and Systems, Rome*, pages 1210–1213, 1982.

[103] J. O'Rourke. The complexity of computing minimum convex covers for polygons. In *Proc. 20th Allerton Conf. Commun. Control Comput.*, pages 75–84, 1982.

[104] J. O'Rourke. The decidability of covering by convex polygons. Report JHU-EECS 82-4, Dept. Elect. Engrg. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1982.

[105] J. O'Rourke. Minimum convex cover for polygons: some counterexamples. Report JHU-EE 82-1, Dept. Elect. Engrg. Comput. Sci., Johns Hopkins Univ., Baltimore, MD, 1982.

[106] J. O'Rourke. Polygon decomposition and switching function minimization. *Comput. Graph. Image Process.*, 18:382–391, 1982.

[107] J. O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, NY, 1987.

[108] J. O'Rourke, C.-B. Chien, T. Olson, and D. Naddor. A new linear algorithm for intersecting convex polygons. *Comput. Graph. Image Process.*, 19:384–391, 1982.

[109] J. O'Rourke and K. J. Supowit. Some NP-hard polygon decomposition problems. *IEEE Trans. Inform. Theory*, IT-30:181–190, 1983.

[110] T. Pavlidis. *Structural Pattern Recognition.* Springer-Verlag, Berlin Heidelberg, 1977.

[111] T. Pavlidis. Survey: A review of algorithms for shape analysis. *Comp. Graphics and Image Processing*, 7:243–258, 1978.

[112] T. Pavlidis and H.-Y. Feng. Shape discrimination. In K. S. Fu, editor, *Syntactic Pattern Recognition*, pages 125–145. Springer Verlag, New York, 1977.

[113] D. Plaisted and J. Hong. A heuristic triangulation algorithm. *Journal of Algorithms*, 8:405–437, 1987.

[114] F. P. Preparata and M. I. Shamos. *Computational Geometry: an Introduction.* Springer-Verlag, New York, NY, 1985.

[115] Arvind Raghunathan. *Polygon Decomposition and Perfect Graphs.* Ph.D. thesis, University of California at Berkeley, Berkeley, California, 1988.

[116] S. Ramaswami, P. Ramos, and G. Toussaint. Converting triangulations to quadrangulations. In *Proc. 7rd Annu. Canadian Conf. on Comput. Geom.*, pages 297–302, 1995.

[117] R. Reckhow. Covering orthogonal convex polygons with three orientations of dents. Report TR87-17, Department of Computing Science, Edmonton, Alberta, 1987.

[118] R. A. Reckhow and J. Culberson. Covering a simple orthogonal polygon with a minimum number of orthogonally convex polygons. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 268–277, 1987.

[119] J.-R. Sack. An $O(n \log n)$ algorithm for decomposing rectilinear polygons into convex quadrilaterals. In *Proc. 20th Allerton Conf. Commun. Control Comput.*, pages 64–75, 1982.

[120] J.-R. Sack. *Rectilinear computational geometry.* Ph.D. thesis, School Comput. Sci., Carleton Univ., Ottawa, ON, 1984. Report SCS-TR-54.

[121] J.-R. Sack and G. T. Toussaint. A linear-time algorithm for decomposing rectilinear polygons into convex quadrilaterals. In *Proc. 19th Allerton Conf. Commun. Control Comput.*, pages 21–30, 1981.

[122] J.-R. Sack and G. T. Toussaint. Guard placement in rectilinear polygons. In G. T. Toussaint, editor, *Computational Morphology*, pages 153–175. North-Holland, Amsterdam, Netherlands, 1988.

[123] B. Schachter. Decomposition of polygons into convex sets. *IEEE Trans. Comput.*, C-27(11):1078–1082, 1978.

[124] D. S. Scott and S. S. Iyengar. Tid - a translation invariant data structure for storing images. *Communications of the ACM*, 29(5):418–429, 1986.

[125] M. Shapira and A. Rappoport. Shape blending using the star-skeleton representation. Manuscript, 1994.

[126] J. Shearer. A class of perfect graphs. *SIAM J. Algebraic Discrete Methods*, 3(3):281–284, 1982.

[127] T. C. Shermer. Covering and guarding polygons using l_k-sets. *Geometriae Dedicata*, 37:183–203, 1991.

[128] T. C. Shermer. Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399, September 1992.

[129] T. C. Shermer. On recognizing unions of two convex polygons and related problems. *Pattern Recognition Letters*, 14(9):737–745, 1993.

[130] V. Soltan and A. Gorpinevich. Minimum dissection of rectilinear polygon with arbitrary holes in to rectangles. *Discrete Comput. Geom.*, 9:57–79, 1993.

[131] S. B. Tor and A. E. Middleditch. Convex decomposition of simple polygons. *ACM Trans. Graph.*, 3:244–265, 1984.

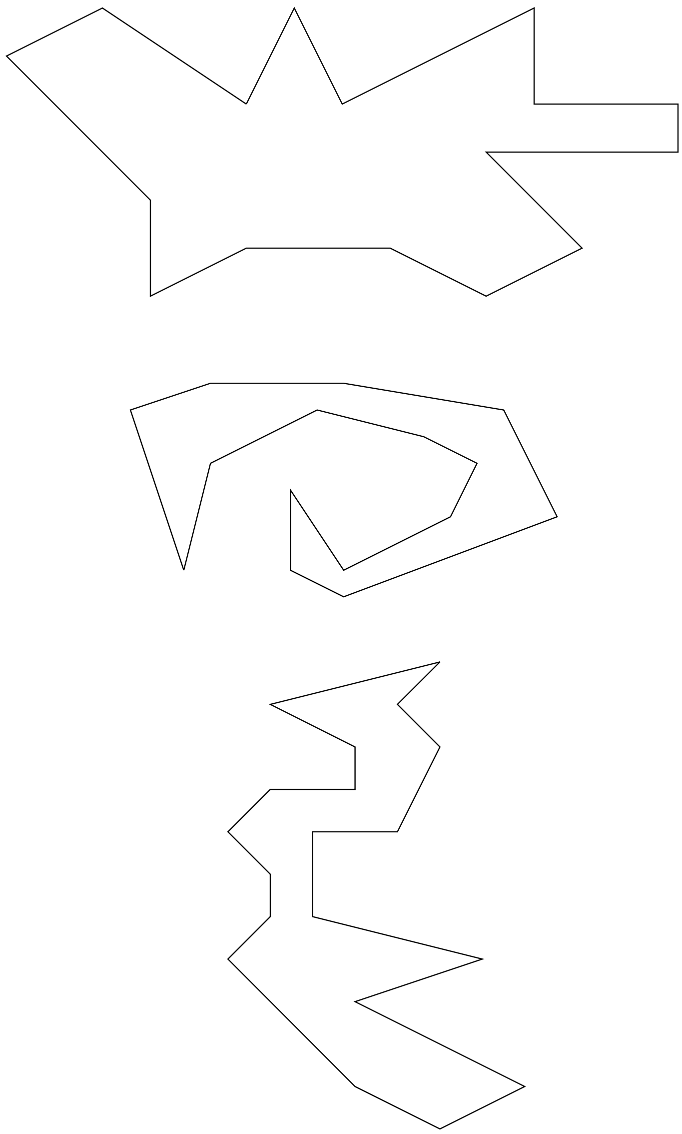
[132] G. Toussaint. Efficient triangulation of simple polygons. *Visual Comput.*, 7:280–295, 1991.

[133] G. T. Toussaint. Pattern recognition and geometrical complexity. In *Proc. Fifth Inter. Conf. on Pattern Recognition*, pages 1324–1347, 1980.

[134] G. T. Toussaint. Quadrangulations of planar sets. In *Proc. 4th Workshop on Algorithms and Data Structures*, pages 218–227, 1995.

[135] A. Y. Wu, S. K. Bhaskar, and A. Rosenfeld. Computation of geometric properties from the medial axis transform in $O(nlogn)$ time. *Computer Vision, Graphics, and Image Processing*, 34:76–92, 1986.

Figure 0.1: (a) a star-shaped polygon, (b) a spiral polygon, and (c) a polygon monotone with respect to the y-axis



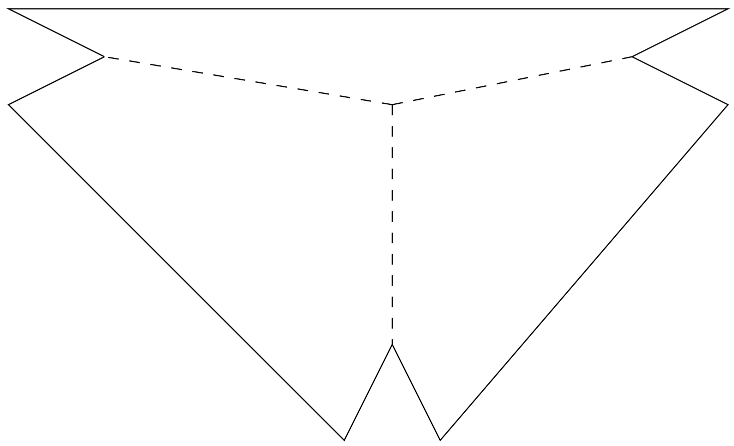
(a)

(b)

(c)

Figure 0.2: A polygon with 9 vertices and 3 reflex vertices partitioned into convex subpolygons (a) with a Steiner point and (b) without Steiner points

(a)



(b)

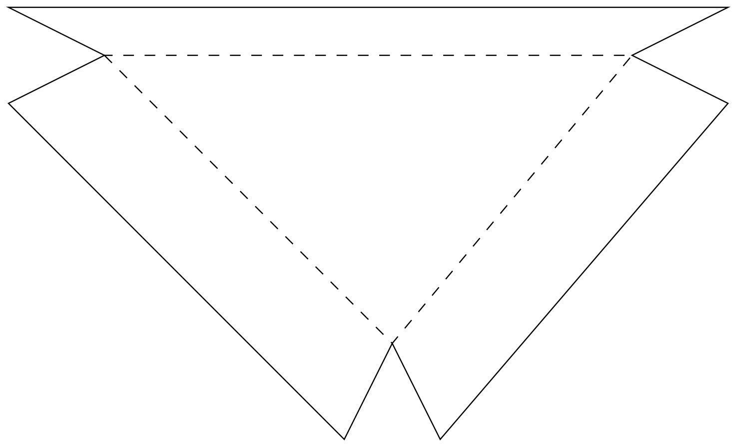


Figure 0.3: An X- pattern

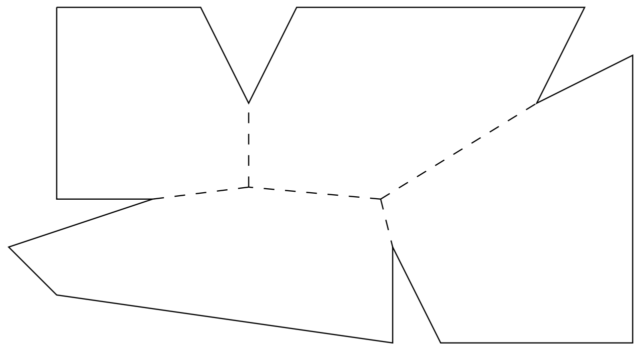


Figure 0.4: A partition into trapezoids

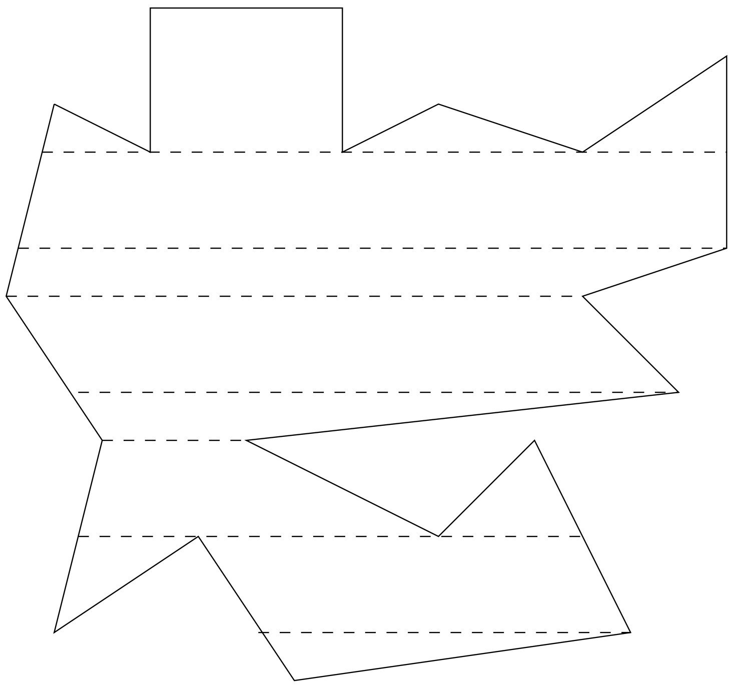


Figure 0.5: Horizontal and vertical chords between reflex vertices

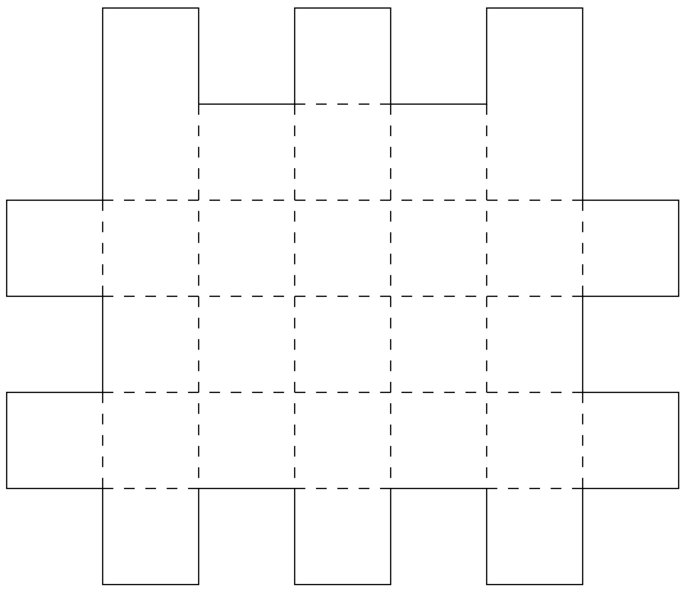
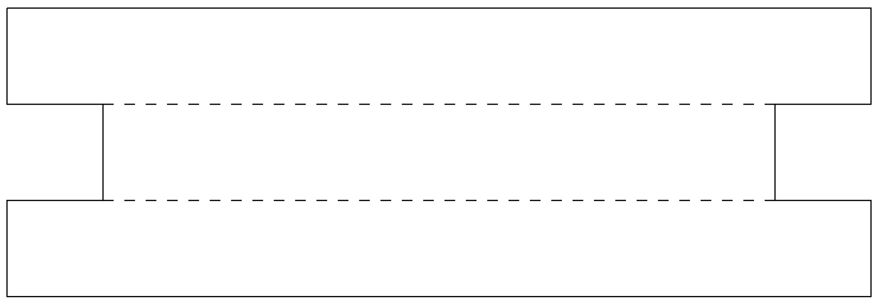


Figure 0.6: A partition using (a) the minimum number of rectangles, and (b) the minimum amount of "ink"

(a)



(b)

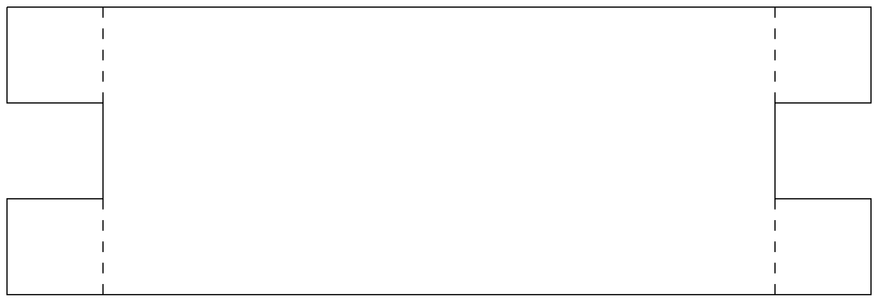


Figure 0.7: A guillotine partition

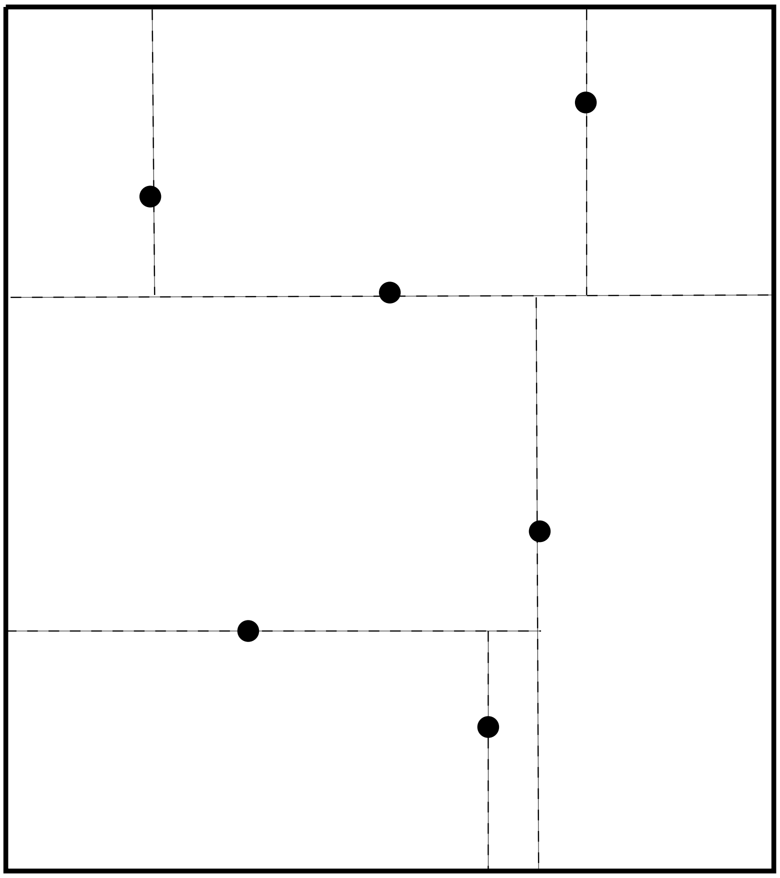


Figure 0.8: Each grid region can be associated with the vertex of a graph

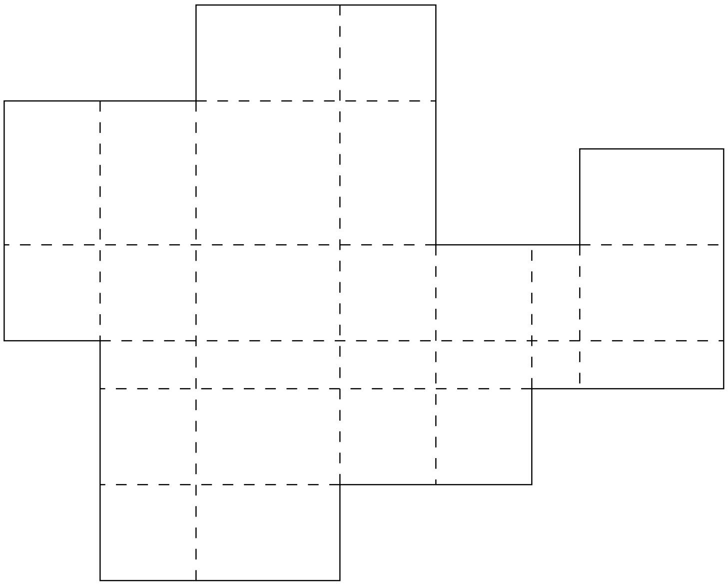


Figure 0.9: A horizontally convex polygon

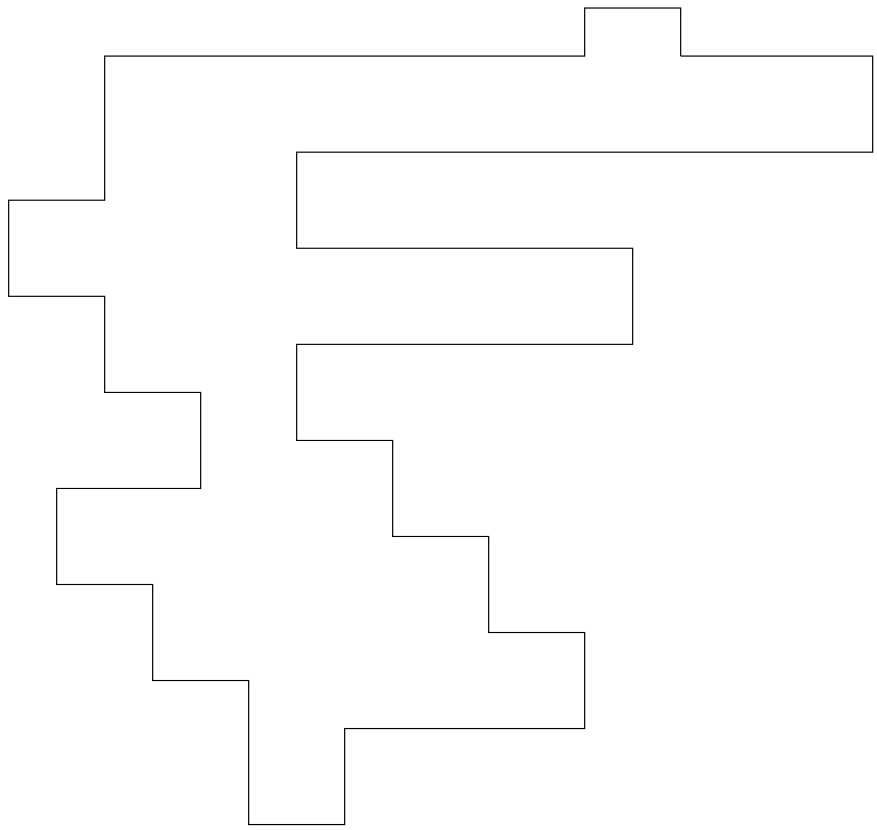


Figure 0.10: A class 3 polygon containing N, W and S dents