DIMACS Technical Report 93-12
March 1993

A Tight Bound for Edge Guards in Rectilinear Monotone Polygons

by

Iliana Bjorling-Sachs[1]
Department of Computer Science
Rutgers University
New Brunswick, New Jersey 08903

[1]Supported by DIMACS research assistantship

DIMACS is a cooperative project of Rutgers University, Princeton University, AT&T Bell Laboratories and Bellcore.

DIMACS is an NSF Science and Technology Center, funded under contract STC–88–09648; and also receives support from the New Jersey Commission on Science and Technology.

ABSTRACT

In this paper we examine a variation of the art gallery problem with each guard allowed to patrol along a designated wall of the gallery and with the gallery itself restricted to the shape of a rectilinear monotone polygon. We show that any such gallery with n vertices is completely covered by at most $\lceil (n-2)/6 \rceil$ guards and that this bound is tight. In addition we give an algorithm for placing the guards, which runs in $O(n)$ time.

1 Introduction

Art gallery problems are problems that ask the following question: how many guards are always sufficient and sometimes necessary to see every point in an art gallery with n walls. The floor plan of the gallery is viewed as a simple polygon. Our knowledge of this polygon may consist only of the fact that n edges are present or we may know that the polygon is of a certain type, e.g. it could be rectilinear or star-shaped. The guards can be stationary or mobile, they may have limited vision or be able to see any distance and in any direction. Stationary guards may be restricted to positions only at the vertices of the polygon or they may be placed at points along the boundary or in the interior of the polygon. Similarly mobile guards may be restrained to move only along the edges of the polygon or they may be allowed to cut across open spaces. The first problem of this type was solved in 1975 by V. Chvatal [3]. He showed that $\lfloor n/3 \rfloor$ stationary guards are always sufficient and sometimes necessary in any simple polygon with n edges. This original art gallery problem and its many variations and extensions are the focus of a book by J. O'Rourke [5] published in 1987. More recent results, obtained since this book was published, can be found in a survey paper by T. Shermer [7].

In this paper we consider a variation of the problem, suggested in 1981 by G. Toussaint, where each guard is allowed to move along one individual edge of the polygon. Guards of this type are called *edge guards*. Two points are *visible* to each other as long as the line segment between them does not intersect the exterior of the polygon. An edge guard *sees* (or *covers*) any point which is visible from at least one point along the guard's edge. The problem is to find a bound on the number of guards necessary to ensure that every point in the polygon is seen by at least one guard. An example of a polygon with n vertices that requires $\lfloor n/4 \rfloor$ edge guards is shown in [5], but no tight bound on the number of required guards exists for a polygon with arbitrary shape. If the polygon is monotone then $\lceil (n-2)/5 \rceil$ edge guards are known to be both necessary and sufficient [2]. Here we give a tight bound of $\lceil (n-2)/6 \rceil$ edge guards for rectilinear monotone polygons. That $\lceil (n-2)/6 \rceil$ edge guards are necessary is shown in section 2 by means of an example. Section 2 also contains the proof of sufficiency in the form of an algorithm that places at most $\lceil (n-2)/6 \rceil$ guards to cover the polygon. An analysis of the algorithm is given in section 3.

2 The tight bound

Let P be a rectilinear polygon with n vertices and assume that P is monotone in the y-direction. The lower bound on the number of edge guards needed to cover every point in P is $\lceil (n-2)/6 \rceil$. The behavior of this bound can be seen in table 1. For $n = 4$ the bound is one and clearly at least one guard must be used in any polygon. The bound increases by one for every six vertices added to the original four. A rectilinear monotone polygon, where one additional guard is necessary for each added group of six vertices is shown in figure 1. This establishes the lower bound.

To show that $\lceil (n-2)/6 \rceil$ is also an upper bound, we present an algorithm that assigns

n	$\lceil (n-2)/6 \rceil$
4	1
6	1
8	1
10	2
12	2
14	2
16	3
18	3
20	3
22	4
...	...

Table 1: The behavior of the lower bound.

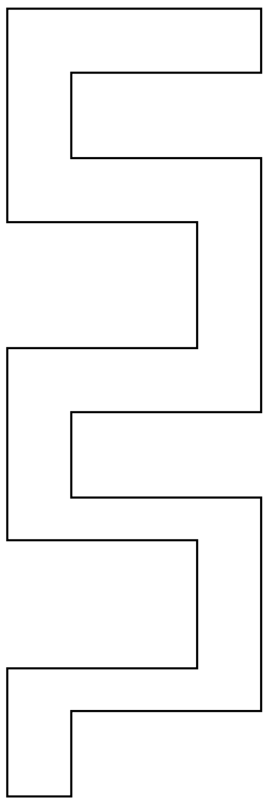


Figure 1: A polygon where each additional six vertices requires one edge guard.

at most this many edge guards to cover the entire polygon. This algorithm is analogous to the one presented in [2] for monotone but non-rectilinear polygons. As a basis for our algorithm, we use Sack's monotone quadrilateralization algorithm [6]. This algorithm starts at the topmost edge and then moves downward creating quadrilaterals by drawing internal diagonals between vertices. We make the following addition to the original algorithm: as the quadrilaterals are formed, they are grouped in such a way that the union of quadrilaterals in each group can be guarded by one edge guard and every group except the last has at least three quadrilaterals in it (see figure 2). The bottommost group may have fewer than

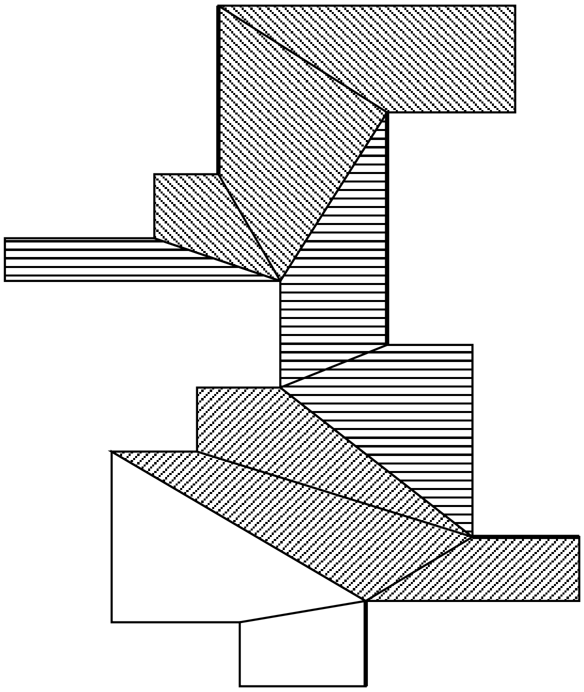


Figure 2: One edge guard, indicated by a heavy edge, is assigned per group.

three quadrilaterals since it will consist of any quadrilaterals left unguarded after the last full group was formed. The partition of P will have $(n-2)/2$ quadrilaterals. Thus there can be at most $(n-2)/6$ complete groups of at least three quadrilaterals and $\lceil (n-2)/6 \rceil$ groups total. Since we assign one edge guard per group, the total number of guards assigned by the algorithm is $\lceil (n-2)/6 \rceil$ and the upper bound is the same as the lower bound.

Before we describe in detail how an edge guard is chosen for each possible configuration of quadrilaterals, we review Sack's quadrilateralization algorithm. This algorithm draws internal diagonals between vertices in such a way that the resulting configurations of quadrilaterals have certain properties that we take advantage of to assign the guards.

In the following an edge is called a *top edge* if the interior of P lies below it and a *bottom edge* if the interior of P lies above it. The *inner* vertex of a horizontal edge is the right endpoint if the edge lies on the left chain and the left endpoint if the edge lies on the right chain. An integral part of the algorithm consists of the quadrilateralization of an *eroded* *pyramid*. A pyramid is a rectilinear polygon which has one bottom edge and is monotone in both the x-direction and the y-direction (see figure 3). An eroded pyramid is a pyramid where the non-vertical edges may be slanted, but monotonicity is maintained. To partition

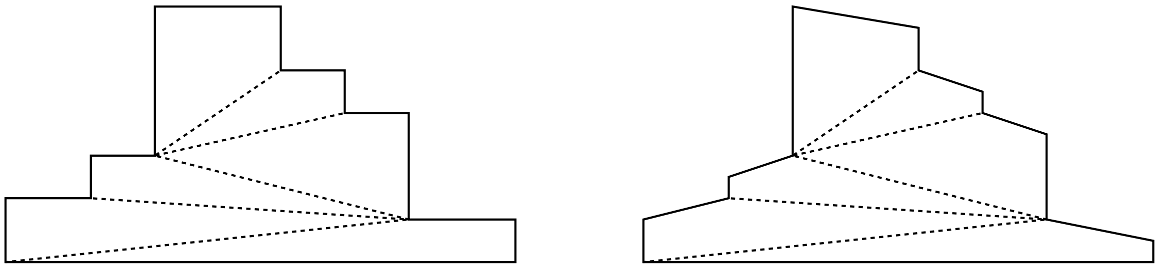


Figure 3: A pyramid, an eroded pyramid and their quadrilateralizations.

an eroded pyramid into quadrilaterals do the following: process the reflex vertices in order of descending y-coordinate and draw one diagonal from each such vertex to the next reflex vertex below it *on the opposite chain*, when no more reflex vertices remain on one chain draw diagonals from the remaining reflex vertices to the vertex with the smallest y-coordinate on the opposite chain. All the diagonals created in this way are *cross-diagonals*. Cross-diagonals stretch across P from one side to the other.

The main algorithm processes the horizontal edges of P from top to bottom. How an edge is treated depends on whether it is a top edge or a bottom edge. Top edges are pushed onto a stack and bottom edges trigger the output of one or more diagonals. When a bottom edge is encountered the stack is popped. There are two possibilities: either the popped edge and the bottom edge have at least one end point each on the same chain or the two edges lie on opposite chains of P. In the first case the inner vertices of the two edges are connected by a diagonal (see figure 4). Such a diagonal is a cross-diagonal if the inner endpoints lie on

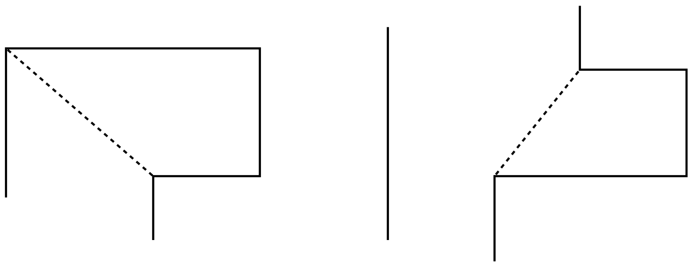


Figure 4: The top edge and the bottom edge have at least one endpoint each on the same chain.

opposite chains and a *side-diagonal* otherwise. Side-diagonal has both endpoints on the same side of P. The new diagonal d is now treated as the next horizontal edge to be processed. Depending on which edge extends further inside P, the still to be partitioned part of P will lie either below or above d and d is considered a top or bottom edge accordingly. In the second case, i.e. if the bottom edge and the popped edge lie on opposite sides of P, two diagonals are drawn (see figure 5). One diagonal between the two left endpoints of the edges

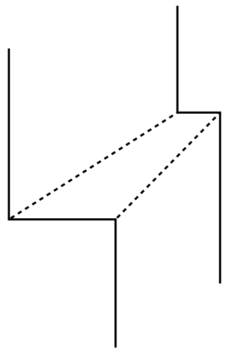


Figure 5: The top edge and the bottom edge lie on opposite chains of P. Two cross-diagonals are generated.

and the other between the two right endpoints. Both these diagonals are cross-diagonals. Let the higher of these diagonals be d_1. Since d_1 stretches across P, the contents of the stack together with d_1 make up a pyramid. Since some of the top edges present on the stack may be side-diagonals that were treated as top edges, the pyramid is an eroded pyramid. At this point the algorithm for quadrilateralization of pyramids is called and the stack is emptied. The lower of the two cross-diagonals is then pushed onto the stack (since the still to be quadrilateralized part of P lies below it, it is treated as a top edge) and the next edge is processed.

As described above two types of diagonals are drawn, cross-diagonals and side-diagonals. The resulting quadrilaterals are of corresponding types. *Side-quadrilaterals* have all vertices on the same chain and can be said to lie either on the right or the left side of P. *Cross-quadrilaterals* have at least one vertex on each chain and stretch across P. We note that side-diagonals are drawn only when a bottom edge and the preceding top edge, in top to bottom order, lie on the same side of P. As soon as a horizontal edge on the opposite chain is encountered, a cross-diagonal is the result. We can make the following observation:

Observation 1 *The extent in the y-direction of a side-quadrilateral or group of adjacent side-quadrilaterals along one chain is spanned by a single vertical edge on the opposite chain.*

We also note that when cross-diagonals are drawn the resulting cross-quadrilaterals are of four possible types (see figure 6). When a pyramid is quadrilateralized, cross-diagonals are drawn between reflex vertices on opposite chains. The top edge of the pyramid is either a cross-diagonal or the extreme top edge of P. The first quadrilateral of the pyramid will consist of this top edge together with two vertical edges and one cross-diagonal drawn between the endpoints of the two vertical edges (see figure 3). We call this a type A quadrilateral. The remaining quadrilaterals in the pyramid all consist of a top edge (original or slanted) followed by a vertical edge on one chain and two cross-diagonals going from this chain to the same vertex on the opposite chain. This type is a type B quadrilateral. It is also possible

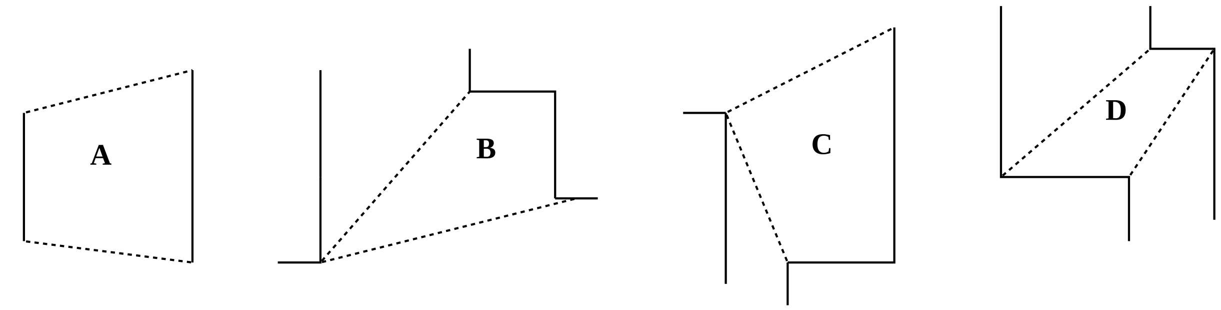


Figure 6: The four types of cross-quadrilaterals. The non-vertical edges can be horizontal edges or side-diagonals.

for a quadrilateral to have a vertical edge followed by a horizontal or slanted bottom edge on one chain, with cross-diagonals going from this chain to one vertex on the opposite chain. This type is a type C quadrilateral and results when a cross-diagonal is drawn between a bottom edge and a popped edge which is a cross-diagonal treated as a top edge. Finally when two edges on opposite sides of P are connected a quadrilateral of type D is created. It consists of two cross-diagonals and two horizontal or slanted edges, one on each side of P.

In our new augmented algorithm we keep a count on the number of currently unguarded quadrilaterals. Side-quadrilaterals are included in the count as soon as a cross-diagonal below them has been drawn. Whenever the algorithm generates a cross-diagonal, we stop and check if the count is at least three, if so then a guard is assigned. Thus each configuration of quadrilaterals, to which a guard is assigned, will contain at least one and at most three cross-quadrilaterals. In the case study below, where every possible configuration of quadrilaterals and its associated guard is presented, the cases are divided into three major categories. Within each category the number of quadrilaterals present is the same. Since side-quadrilaterals are included in the count only when the cross-diagonal below them is generated, it is possible that the count never is exactly three. Instead the count may jump from two to q, where $q-1$ side-quadrilaterals were included when the last cross-diagonal was generated. It may be necessary to exclude some of these side-quadrilaterals from the group being formed and instead leave them unguarded until the next stop. When this is the case, we make sure that the guard is chosen in such a way that the following invariant holds:

Invariant: *Following the stop at a cross-diagonal any side-quadrilaterals left unguarded by the newly assigned guard lie immediately above and on the same side as the bottom-most vertex of that cross-diagonal.*

2.1 Exactly one cross-quadrilateral is present

Assume that a cross-diagonal has just been generated and upon checking we find that the count is at least three. Of the unguarded quadrilaterals, exactly one is a cross-quadrilateral. The remaining quadrilaterals (of which there are at least two) are side-quadrilaterals. As stated above the cross-quadrilateral must be of type A, B, C or D. The four cases are treated separately.

Case 2.1.1 *The single cross-quadrilateral present is of type A.*

A quadrilateral of type A has two vertical edges on opposite sides of P (see figure 7). These edges are connected by cross-diagonals. Let the endpoints be l_i and l_j on the left side

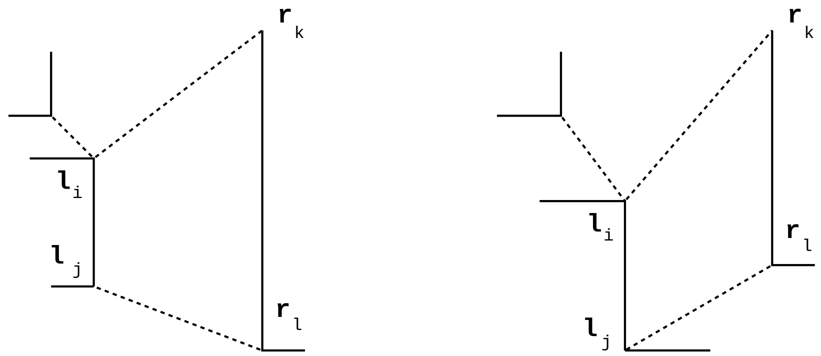


Figure 7: One cross-quadrilateral of type A is present.

and r_k and r_l on the right side. Assume that l_i lies below r_k. A symmetric case ensues if l_i lies above r_k. At least two side-quadrilaterals must be present to make the count three or we would not have stopped to assign a guard. Since l_il_j and r_kr_l are real edges of P, no side-quadrilateral lies between the two diagonals l_ir_k and l_jr_l. The present side-quadrilaterals must have been left unguarded when the previous guard was assigned. As stated in the invariant they lie immediately above the bottom-most vertex of the diagonal l_ir_k, in this case vertex l_i. From observation 1 we know that these side-quadrilaterals are spanned by a vertical edge on the right chain. Since l_i lies below r_k, the spanning edge is r_kr_l and a guard on this edge sees all quadrilaterals in the configuration. $\diamond$

Case 2.1.2 *The single cross-quadrilateral present is of type B.*

Quadrilaterals of type B have a single vertex, which is the bottom endpoint of a vertical edge, on one chain and a horizontal or slanted top edge followed by a vertical edge on the opposite chain (see figure 8). Assume that the single vertex l_i lies on the left chain and the three vertices on the right chain are in order r_j, r_k and r_l. A type B quadrilateral is created when a pyramid is partitioned. Since exactly one diagonal is drawn from each reflex vertex in the pyramid to a vertex below it, no vertex of the quadrilateral is the top endpoint of more than one diagonal. Vertex l_i cannot lie above vertex r_j or both l_ir_j and l_ir_l would slope downward from l_i. Furthermore, l_i cannot lie between r_j and r_k since the side-quadrilaterals between these two vertices are spanned by a single edge on the left chain. Thus vertex l_i must lie between vertices r_k and r_l or below vertex r_l. At least two side-quadrilaterals are be present between r_j and r_k. These side-quadrilaterals are spanned by the edge $l_{i-1}l_i$ and a guard on this edge sees all the side-quadrilaterals in addition to the cross-quadrilateral. $\diamond$

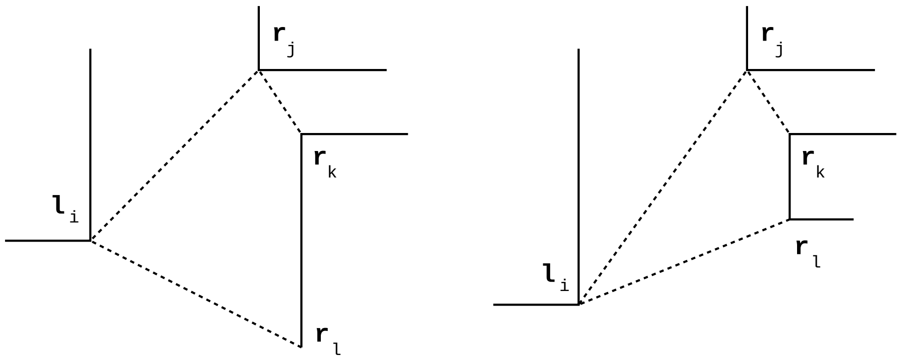


Figure 8: One cross-quadrilateral of type B is present.

Case 2.1.3 *The single cross-quadrilateral present is of type C.*

This type of quadrilateral has a single vertex, which is the top endpoint of a vertical edge, on one side of the polygon and a vertical edge followed by a horizontal or slanted bottom edge on the opposite side of the polygon. Assume that the single vertex l_i lies on the left side and that the three vertices on the right side are in order r_j, r_k and r_l (see figure 9). This type of quadrilateral was created when inner vertex r_l of the bottom edge $r_k r_l$ was

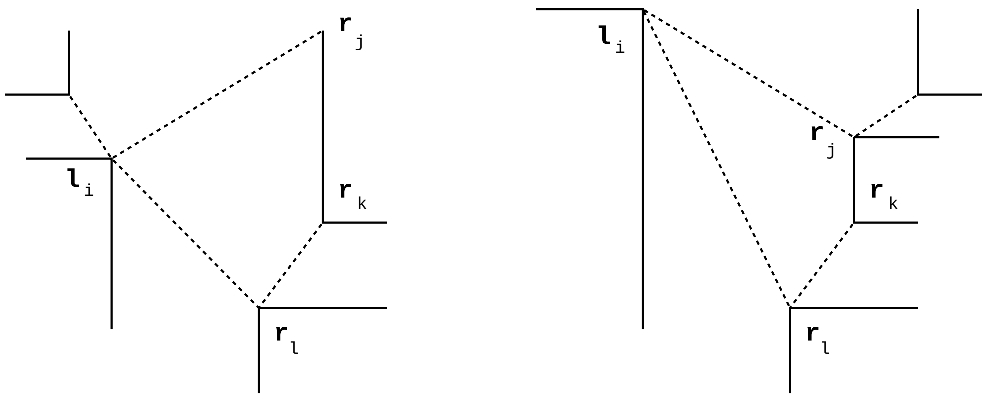


Figure 9: One cross-quadrilateral of type C is present.

connected to vertex l_i of a top edge (slanted or original), which stretches across P. The top edge was popped from the stack and the bottom edge is the edge currently being processed. Thus vertex l_i lies above vertices r_k and r_l, either between r_j and r_k or above r_j. If l_i lies between r_j and r_k and l_i is the bottom endpoint of a side-diagonal then unguarded side-quadrilaterals may be present on the left side immediately above l_i. If l_i lies above r_j then unguarded side-quadrilaterals may lie on the right side immediately above r_j. Any number of side-quadrilaterals may also be present between r_k and r_l. If two or more side-quadrilaterals are present above l_i, then edge $r_j r_k$ spans these quadrilaterals and a guard on this edge sees said side-quadrilaterals and the cross-quadrilateral. Any side-quadrilaterals between r_k and

r_l are left unguarded for the next guard assignment. If no more than one side-quadrilateral is present above l_i or if all side-quadrilaterals lie on the right side, then a guard on edge $l_i l_{i+1}$ sees all the quadrilaterals present. $\diamond$

Case 2.1.4 *The single cross-quadrilateral present is of type D.*

A quadrilateral of type D has two horizontal or slanted edges on opposite chains of P connected by cross-diagonals (see figure 10). One of the horizontal edges is a bottom edge and

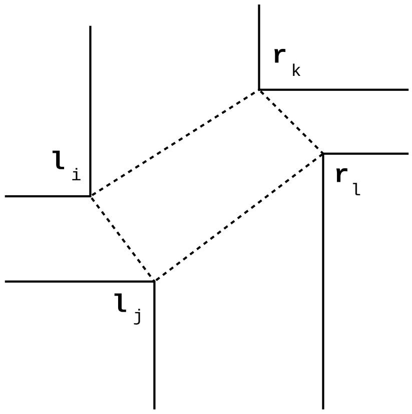


Figure 10: One cross-quadrilateral of type D is present.

the other is a top edge. Assume that the bottom edge lies on the left side and has endpoints l_i and l_j. Let the endpoints of the top edge on the right side be r_k and r_l. Since both these edges may in fact be side-diagonals treated as horizontal edges, side-quadrilaterals may be present on both the left and the right side. If at least two side-quadrilaterals lie between r_k and r_l, then a guard on edge $l_{i-1}l_i$ sees these in addition to the cross-quadrilateral. Any side-quadrilaterals between l_i and l_j are left for the next guard assignment. If at most one side-quadrilateral lies between r_k and r_l, then a guard on edge $r_l r_{l+1}$ sees this side-quadrilateral, the cross-quadrilateral and all side-quadrilaterals between l_i and l_j. $\diamond$

These four cases and their symmetric counterparts exhaust the possible configurations in the category where exactly one cross-quadrilateral is present.

2.2 Exactly two cross-quadrilaterals are present

Next we consider the category where exactly two cross-quadrilaterals are present in a configuration. Because the edges of P are alternately vertical and horizontal, only certain combinations of adjacent cross-quadrilaterals are possible. Consider first the possibility that the top cross-quadrilateral Q_1 is of type A The two bottom vertices of this type of quadrilateral are bottom endpoints of vertical edges (see figure 7). Cross-quadrilateral Q_2, which lies immediately below Q_1, cannot have a top vertex which is the top endpoint of a vertical edge, since this would mean that there were two consecutive vertical edges on P's boundary. Thus Q_2 cannot be of type A or C but instead must be of type B or D. The two possibilities are treated as separate cases.

Case 2.2.1 *Exactly two cross-quadrilaterals are present. The top quadrilateral is of type A and the quadrilateral below is of type B.*

Let the top quadrilateral Q_1 have vertices l_i and l_j on the left side and vertices r_k and r_l on the right side (see figure 11). Assume vertex l_i lies below vertex r_k. Symmetric cases

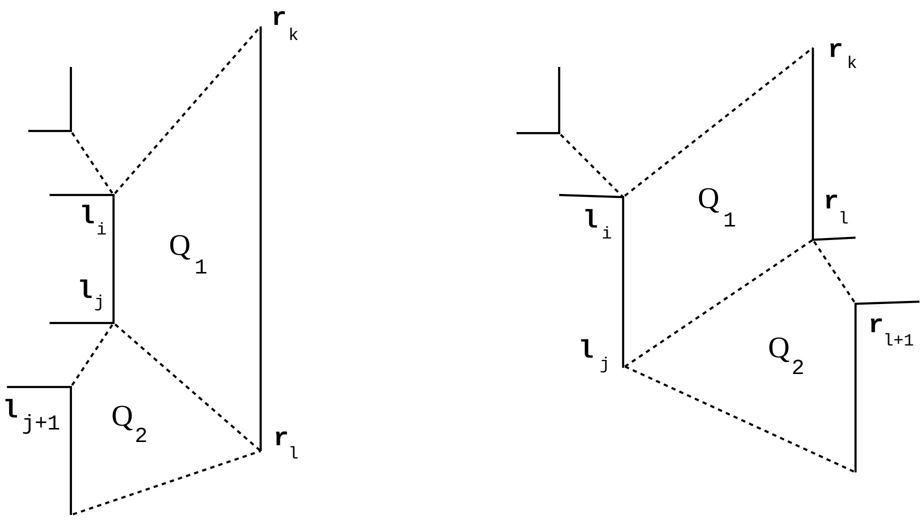


Figure 11: The top cross-quadrilateral is of type A, the one below is of type B.

ensue when vertex l_i lies above r_k. With l_i below r_k any side-quadrilateral left unguarded from the previous guard assignment must lie on the left side. Note that at most one such side-quadrilateral exists, since if there were two or more, the count would have reached three when diagonal $l_j r_l$ was generated. It is possible for vertex l_j to lie either above or below vertex r_l. In the first case, the second quadrilateral Q_2 has a single vertex on the right side. This is the vertex r_l. Any number of side-quadrilaterals may be present between vertices l_j and l_{j+1}, but these are spanned by the edge $r_k r_l$ and a guard on this edge sees all the quadrilaterals involved. If vertex l_j instead lies below vertex r_l then Q_2 has a single vertex on the left side. This is the vertex l_j. Side-quadrilaterals may lie between r_l and r_{l+1}, but these are spanned by edge $l_i l_j$ and a guard on this edge sees all quadrilaterals. $\diamond$

Case 2.2.2 *Exactly two cross-quadrilaterals are present. The top quadrilateral is of type A and the quadrilateral below is of type D.*

Let the top quadrilateral Q_1 have vertices l_i and l_j on the left side and vertices r_l and r_m on the right side (see figure 12). Assume l_i lies below r_l. The vertices of quadrilateral Q_2 are l_j and l_k on the left side and r_m and r_n on the right side. As in the previous case a single

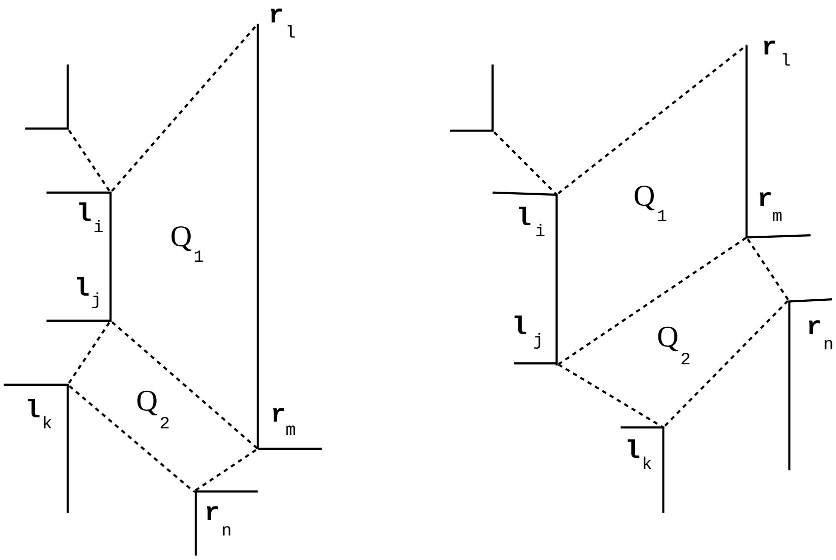


Figure 12: The top cross-quadrilateral is of type A, the one below is of type D.

unguarded side-quadrilateral can lie immediately above l_i on the left side. Any number of side-quadrilaterals are possible between vertices l_j and l_k on the left side and vertices r_l and r_m on the right side. If vertex l_j lies above vertex r_m then a guard on edge $r_l r_m$ sees all side-quadrilaterals on the left side, the two cross-quadrilaterals and one side-quadrilateral on the right side. Additional side-quadrilaterals on the right side are left for the next guard assignment. If vertex l_j lies below vertex r_m then a guard on edge $l_i l_j$ sees the single side-quadrilateral above l_i, all side-quadrilaterals on the right side, the two cross-quadrilaterals and one of the side-quadrilaterals on the left side below l_j. Any additional side-quadrilaterals below l_j are left for the next guard assignment. $\diamond$

If instead the top quadrilateral is of type B then it is again true the two bottom vertices of Q_1 are bottom endpoints of vertical edges and using the same argument as above Q_2 must be of type B or D.

Case 2.2.3 *Exactly two cross-quadrilaterals, both of type B, are present.*

Assume that Q_1 has a single vertex l_i on the left side and vertices r_j, r_k and r_l on the right side (see figure 13). At most one side-quadrilateral is present between r_j and r_k, since if there were more then the count would have reached three when cross-diagonal $l_i r_l$ was generated. As described in case 2.1.2, the lone vertex on one side of a type B quadrilateral is the top endpoint of exactly one cross-diagonal. Thus if l_i lies between r_k and r_l then the cross-quadrilateral Q_2 below Q_1 has its single vertex on the right side, or l_i would be the top endpoint of two cross-diagonals. Similarly, if l_i lies below r_l then Q_2 has its single vertex on the left side or r_l would be the top endpoint of two cross-diagonals. In the first case a guard on edge $r_k r_l$ sees the single side-quadrilateral that may lie above r_k, the two

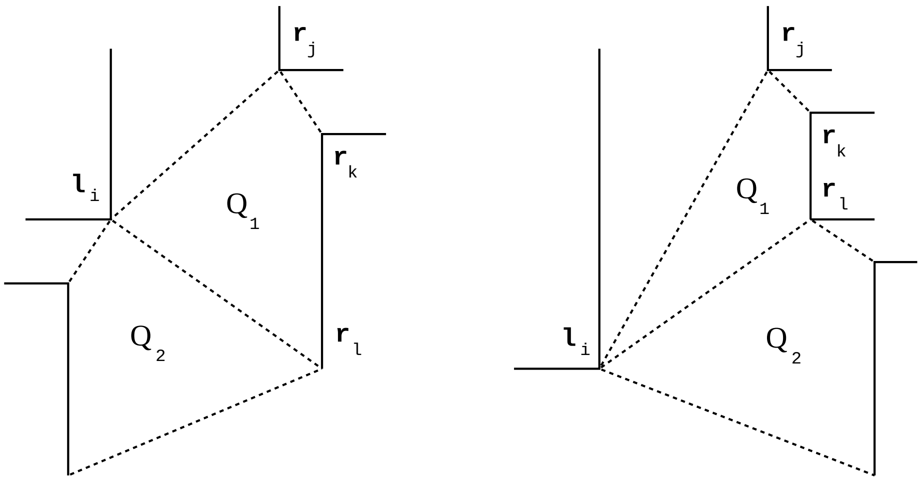


Figure 13: Both cross-quadrilaterals are of type B.

cross-quadrilaterals and any side-quadrilaterals present between l_i and l_{i+1}. In the second case, i.e. if l_i lies below r_l, a guard on edge $l_{i-1}l_i$ sees all quadrilaterals present. $\diamond$

Case 2.2.4 *Exactly two cross-quadrilaterals are present. The top quadrilateral is of type B and the quadrilateral below is of type D.*

Assume that Q_1 has a single vertex l_i on the left side and vertices r_k, r_l and r_m on the right side (see figure 14). The vertices of Q_2 are l_i and l_j on the left side and r_m and r_n on the right

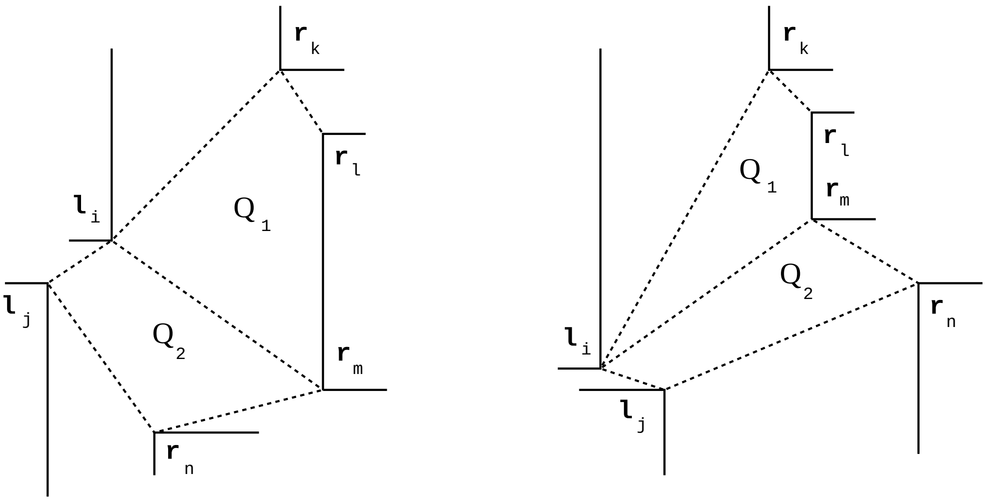


Figure 14: The top cross-quadrilateral is of type B, the one below is of type D.

side. In addition to the single side-quadrilateral that may lie between r_k and r_l, any number of side-quadrilaterals may lie between l_i and l_j and also between r_m and r_n. If vertex l_i lies between r_l and r_m then a guard on edge r_lr_m sees the single side-quadrilateral that may be present above r_l, the two cross-quadrilaterals, all side-quadrilaterals between l_i and l_j and at least one of the side-quadrilaterals between r_m and r_n. Any additional side-quadrilaterals below r_m are left for the next guard assignment. If vertex l_i lies below r_n then a guard on edge $l_{i-1}l_i$ sees all side-quadrilaterals on the right side, the two cross-quadrilaterals and at least one side-quadrilateral between l_i and l_j. Additional side-quadrilaterals below l_i are left unguarded for the next guard assignment. $\diamond$

Next let Q_1 be of type C. The bottom cross-diagonal of this type of quadrilateral was drawn when the inner endpoints of two horizontal edges (slanted or original) were connected. The edges following below each of these horizontal edges are vertical edges. Thus the two endpoints of this diagonal, which are also the top vertices of Q_2, are top endpoints of vertical edges of P and Q_2 must be of type C or type A.

Case 2.2.5 *Exactly two cross-quadrilaterals, both of type C are present.*

Assume that Q_1 has a single vertex l_i on the left chain and three vertices r_l, r_m and r_n on the right chain (see figure 15). No more than one side-quadrilateral can be present above l_ir_n or the count would have been three when the cross-diagonal was generated. If l_i lies below r_l then this side-quadrilateral lies either immediately above l_i on the left side or between r_m and r_n and if l_i lies above r_l then the side-quadrilateral lies above r_l on the right side or between r_m and r_n.. If Q_2 also has a single vertex on the left chain then a guard on

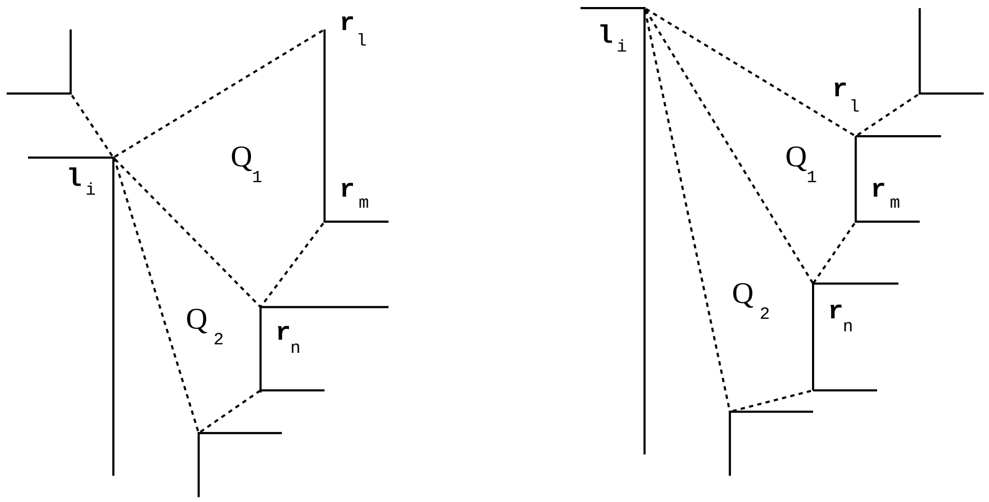


Figure 15: Both cross-quadrilaterals are of type C and both have a single vertex on the left side.

edge l_il_{i+1} sees all quadrilaterals. If instead Q_2 has a single vertex on the right chain (see figure 16) and vertices l_i, l_j and l_k on the left chain then a guard on edge l_il_j sees the single side-quadrilateral that may be present above l_ir_n, the two cross-quadrilaterals and at least one side-quadrilateral between l_j and l_k. Any additional side-quadrilaterals between l_j and l_k are left for the next guard assignment. $\diamond$

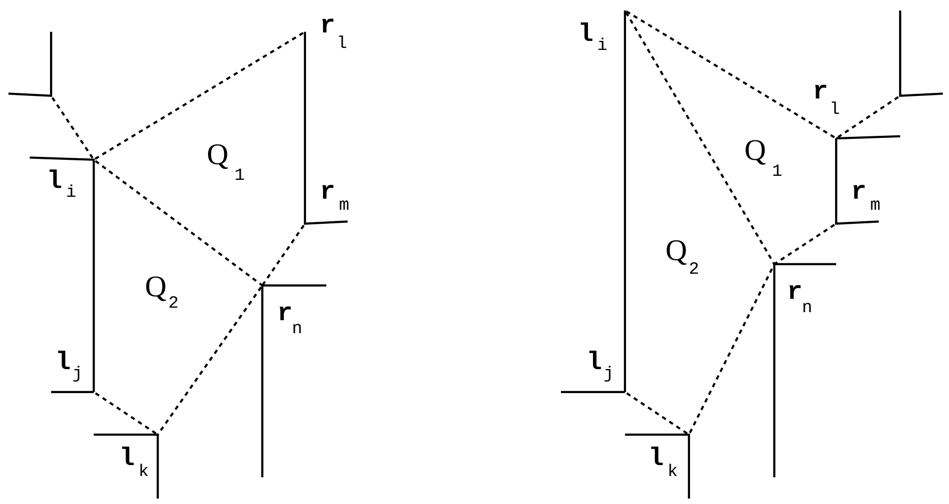


Figure 16: Both cross-quadrilaterals are of type C. They have single vertices on opposite sides of P.

Case 2.2.6 *Exactly two cross-quadrilaterals are present. The top quadrilateral is of type C and the quadrilateral below is of type A.*

Assume Q_1 has vertex l_i on the left chain and vertices r_k, r_l and r_m on the right chain. Q_2 has vertices l_i and l_j on the left chain and vertices r_m and r_n on the right chain (see figure 17) Vertex l_i may lie above r_k or between r_k and r_l. In either case a guard on edge $l_i l_j$ sees

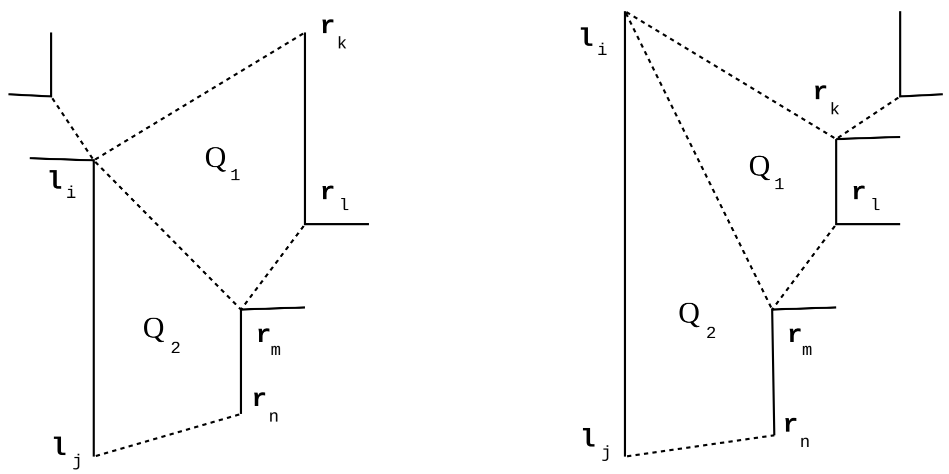


Figure 17: The top cross-quadrilateral is of type C, the one below is of type A.

all the quadrilaterals involved. $\diamond$

Finally we consider the possibility that Q_1 is of type D. A type D cross-quadrilateral is created when one cross-diagonal is drawn between the left endpoints of two horizontal edges on opposite chains and a second cross-diagonal is drawn between the right endpoints of the two same edges. The bottom vertices of Q_1, which are also the top vertices of Q_2, are then top endpoints of vertical edges of P and Q_2 must be of type C or type A.

Case 2.2.7 *Exactly two cross-quadrilaterals are present. The top quadrilateral is of type D and the quadrilateral below is of type A.*

Let Q_1 have vertices l_i and l_j on the left chain and vertices r_k and r_l on the right chain. Assume that $l_i l_j$ is the bottom edge (slanted or original) and $r_k r_l$ is the top edge (slanted or original). Then l_i lies below r_l (see figure 18). Exactly one side-quadrilateral is present. If

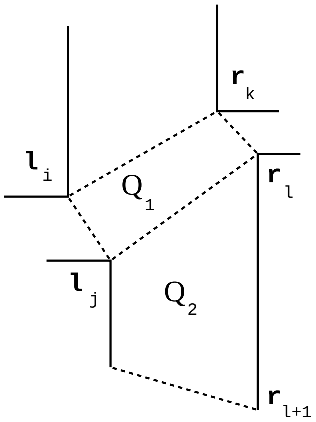


Figure 18: The top cross-quadrilateral is of type D, the one below is of type A.

there were more than one then the count would have reached three when cross-diagonal $l_j r_l$ was generated and we would have stopped before creating Q_2. If no side-quadrilateral was present the count would only be two after the creation of Q_2 and we would not have stopped to assign a guard. The single side-quadrilateral lies either between l_i and l_j or between r_k and r_l. In either case a guard on edge $r_l r_{l+1}$ sees all three quadrilaterals. $\diamond$

Case 2.2.8 *Exactly two cross-quadrilaterals are present. The top quadrilateral is of type D and the quadrilateral below is of type C.*

Assume, as for the previous case, that Q_1 has a bottom edge $l_i l_j$ on the left side and a top edge $r_k r_l$ on the right side. Q_2 has a single vertex on one side and three vertices on the opposite side. It is possible for the single vertex to lie on either side of P (see figure 19). If the side chosen for Q_2's single vertex is the left side then a guard on edge $r_l r_{l+1}$ sees the one side-triangle that may be present above $l_j r_l$, the two cross-quadrilaterals and one side-quadrilateral below r_l. Any additional side-quadrilaterals below r_l are left for the next guard assignment. If instead the side chosen for the single vertex is the right side then the same guard sees all the present quadrilaterals. $\diamond$

The above eight cases and their symmetric counterparts exhaust the possible configurations in the category where exactly two cross-quadrilaterals are present.

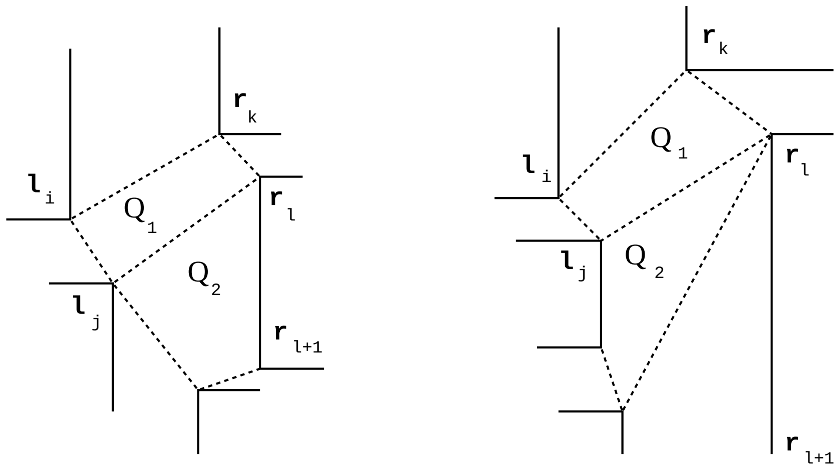


Figure 19: The top cross-quadrilateral is of type D, the one below is of type C.

2.3 Exactly three cross-quadrilaterals are present

The third category of cases is the one where exactly three cross-quadrilaterals are present in a configuration. Let these cross-quadrilaterals be Q_1, Q_2 and Q_3 in top to bottom order. Since we did not stop to assign a guard after Q_2 was created, the count at that point must have been exactly two and any side-quadrilaterals present in the configuration lie below Q_2. Consider first the possibility that Q_1 is of type A. We know from the discussion of configurations where exactly two cross-quadrilaterals are present, that Q_2 must be of type B or type D. Similarly if Q_2 is of type B then Q_3 is either of type B or type D and if Q_2 is of type D then Q_3 must be either of type C or type A. Thus the possible combinations of types of three consecutive cross-quadrilaterals, where the top quadrilateral is of type A, are ABB, ABD, ADC and ADA. Each combination is treated as a separate case.

Case 2.3.1 *Exactly three cross-quadrilaterals are present. They are, in order from top to bottom, of types A, B and B.*

Let Q_1 have vertices l_i and l_j on the left side and vertices r_l and r_m on the right side. Assume that l_i lies below r_m (see figure 20). Symmetric cases ensue when l_i lies above r_m. Q_2 is a quadrilateral of type B, so one of its four vertices lies on one chain and the remaining three vertices lie on the opposite chain. As explained in case 2.2.1, Q_2's isolated vertex must lie on the left side and that vertex is l_j. On the right side Q_2 has a horizontal edge $r_m r_n$ followed by a vertical edge $r_n r_p$. The third cross-quadrilateral Q_3 is also of type B. If l_j lies below r_p then Q_3 has the single vertex l_j on the left side and a horizontal edge $r_p r_q$ followed by a vertical edge on the right side. Since no side-quadrilaterals can lie above Q_3 any present side-quadrilaterals lie between r_p and r_q and are spanned by $l_i l_j$. A guard on this edge sees all side-quadrilaterals and the three cross-quadrilaterals. If l_j lies above r_p then Q_3 has the single vertex r_p on the right side and a horizontal edge $l_j l_k$ followed by a vertical edge on the left side. Any side-quadrilaterals lie between l_j and l_k. A guard on edge $l_i l_j$ sees

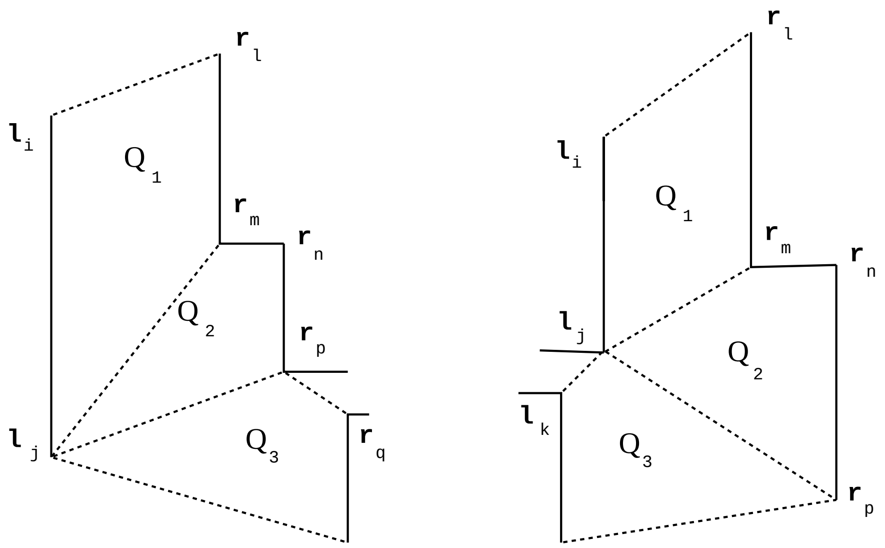


Figure 20: The top cross-quadrilateral is of type A, the other two of type B.

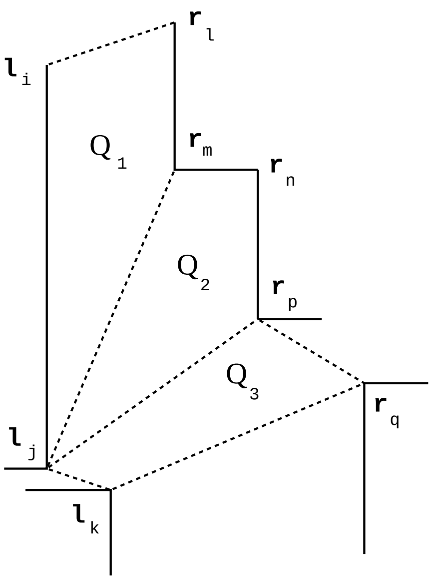
one such side-quadrilateral and all cross-quadrilaterals. If two or more side-quadrilaterals are present, the newly formed group of quadrilaterals will consist of Q_1, Q_2 and one of the side-quadrilaterals. Q_3 and the remaining side-quadrilaterals will belong to the following group. ◇

Case 2.3.2 *Exactly three cross-quadrilaterals are present. They are, in order from top to bottom, of types A, B and D.*

Let Q_1 and Q_2 be as described in case 2.3.1 above (see figure 21). The third cross-quadrilateral has a bottom edge on one chain connected by cross-diagonals to a top edge on the opposite chain. If l_j lies below r_p then Q_3 has bottom edge l_jl_k on the left side and top edge r_pr_q on the right side. Side-quadrilaterals may lie between l_j and l_k and also between r_p and r_q. A guard on edge l_il_j sees all side-quadrilaterals on the right side and the three cross-quadrilaterals. Any side-quadrilaterals on the left side are left for the next guard assignment. If instead l_j lies above r_p then l_jl_k is the top edge of Q_3 and r_pr_q is the bottom edge. A guard on edge l_il_j sees the three cross-quadrilaterals and one side-quadrilateral between l_j and l_k. If more than one side-quadrilateral is present between l_i and l_j then the newly formed group consists of Q_1Q_2 and one side-quadrilateral. The remaining side-quadrilaterals and Q_3 will belong to the next group. ◇

Case 2.3.3 *Exactly three cross-quadrilaterals are present. They are, in order from top to bottom, of types A, D and C.*

Let Q_1 have the vertices l_i and l_j on the left side and r_l and r_m on the right side (see figure 22). Assume that l_i lies below r_m. Symmetric cases ensue when l_j lies above r_m. Q_2 has vertices l_j and l_k on the left side and vertices r_m and r_n on the right side. Note that



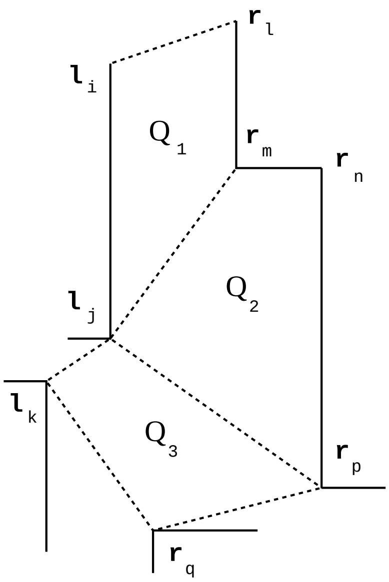
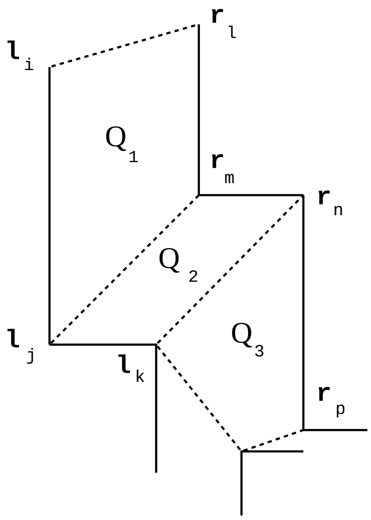


Figure 21: The top cross-quadrilateral is of type A, the middle of type B and the bottom of type D.



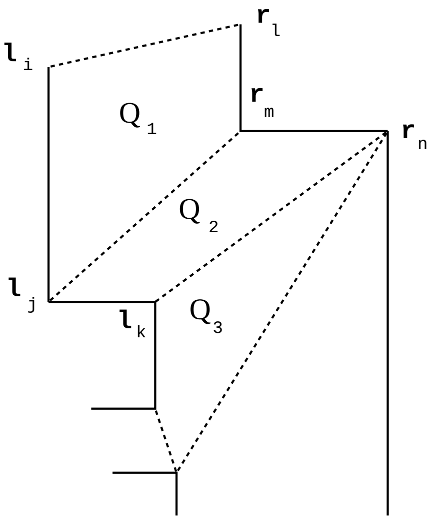


Figure 22: The top cross-quadrilateral is of type A, the middle of type D and the bottom of type C.

since no side-quadrilateral is present above Q_3, both l_jl_k and r_mr_n are original horizontal edges of P. Q_3 has a vertical edge followed by a bottom edge on either the left or the right side. This bottom edge may be a side-diagonal with side-quadrilaterals present behind it. A guard on edge l_jl_k sees the three cross-quadrilaterals. All side-quadrilaterals are left for the next guard assignment. $\diamond$

Case 2.3.4 *Exactly three cross-quadrilaterals are present. They are, in order from top to bottom, of types A, D and A.*

Let Q_1 and Q_2 be as described in case 2.3.3 above (see figure 23). Since Q_3 is of type A,

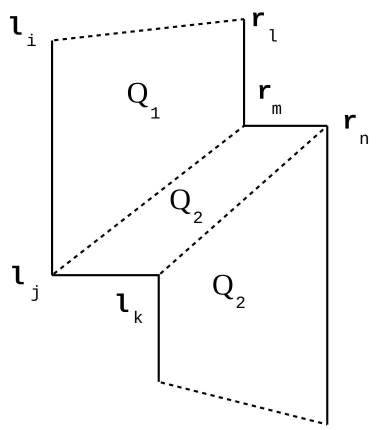


Figure 23: The top and bottom cross-quadrilaterals are of type A, the middle is of type D.

no side-quadrilaterals are present and a guard on edge l_jl_k sees all quadrilaterals. $\diamond$

When the top quadrilateral is instead of type B, we know from section 2.2 that Q_2 must be of either type B or type D. With Q_2 of type B, Q_3 is of type B or D and with Q_2 of type D, Q_3 is of type C or A. Thus there are four possible combinations of types when Q_1 is of type B. They are BBB, BBD, BDC and BDA.

Case 2.3.5 *Exactly three cross-quadrilaterals are present. All three are of type B.*

A type B quadrilateral has one vertex on one side of P and three vertices on the opposite side. Assume that Q_1 has a single vertex l_i on the left side and vertices r_l, r_m and r_n on the right side. We know from case 2.2.3 (see figure 13), that depending on whether l_i lies above or below r_n, Q_2 will have its single vertex on the right or the left side. First consider the case when l_i lies above r_n (see figure 24) and let Q_2 have vertices l_i, l_j and l_k on the left side and vertex r_n on the right side. Using the same argument again, we see that if r_n lies above l_k then Q_3 has one vertex on the left side and three vertices on the right side and if r_n lies below l_k then Q_3 has three vertices on the left side and one on the right side. If side-quadrilaterals are present in the configuration, they lie behind the slanted top edge of Q_3. If r_n lies above

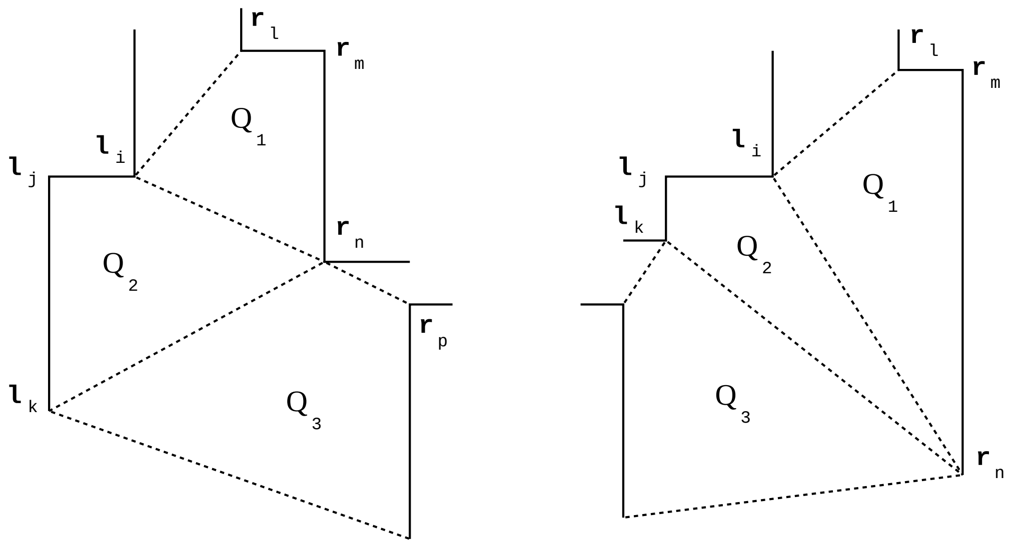


Figure 24: All three cross-quadrilaterals are of type B, the top two have a single vertex on the left side.

l_k then this top edge is edge r_nr_p on the right side of P. A guard on edge r_mr_n sees one side-quadrilateral and all cross-quadrilaterals. If two or more side-quadrilaterals are present then the newly formed group will consist of one such side-quadrilateral and the two top cross-quadrilaterals. The remaining side-quadrilaterals and Q_3 will belong to the following group. If instead r_n lies below l_k then the side-quadrilaterals lie on the left side and are spanned by edge r_mr_n. A guard on this edge sees all quadrilaterals. Next consider the case when l_i lies below r_n (see figure 25) and let Q_2 has vertex l_i on the left side and vertices r_n, r_p and r_q on the right side. If l_i lies above r_q then any present side-quadrilaterals lie on the left side. A guard on edge $l_{i-1}l_i$ sees one side-quadrilateral and all cross-quadrilaterals. If more than one side-quadrilateral is present the new group will consist of Q_1, Q_2 and one side-quadrilateral. The remaining side-quadrilaterals and Q_3 will belong to the next group. If l_i lies below r_q then the side-quadrilaterals lie on the right side and are spanned by edge $l_{i-1}l_i$. A guard on this edge sees all quadrilaterals. $\diamond$

Case 2.3.6 *Exactly three cross-quadrilaterals are present. They are of types B, B and D.*

As for case 2.3.5, the orientation of Q_2 depends on the position of l_i relative to r_n. Consider first the case that l_i lies above r_n (see figure 26). Q_2 has vertices l_il_j and l_k on the left side and vertex r_n on the right side. If r_n lies above l_k then Q_3 has a top edge r_nr_p on the right side and a bottom edge on the left side. Side-quadrilaterals may be present behind both theses edges. A guard on edge r_mr_n sees one side-quadrilateral between r_n and r_p and all cross-quadrilaterals. Any side-quadrilaterals below l_k are left for the next guard assignment. If more than one side-quadrilateral lies between r_n and r_p then the new group will consist of one of these side-quadrilaterals, Q_1 and Q_2. Q_3 and all remaining side-quadrilaterals will belong to the next group. If r_n lies below l_k then the top edge of Q_3

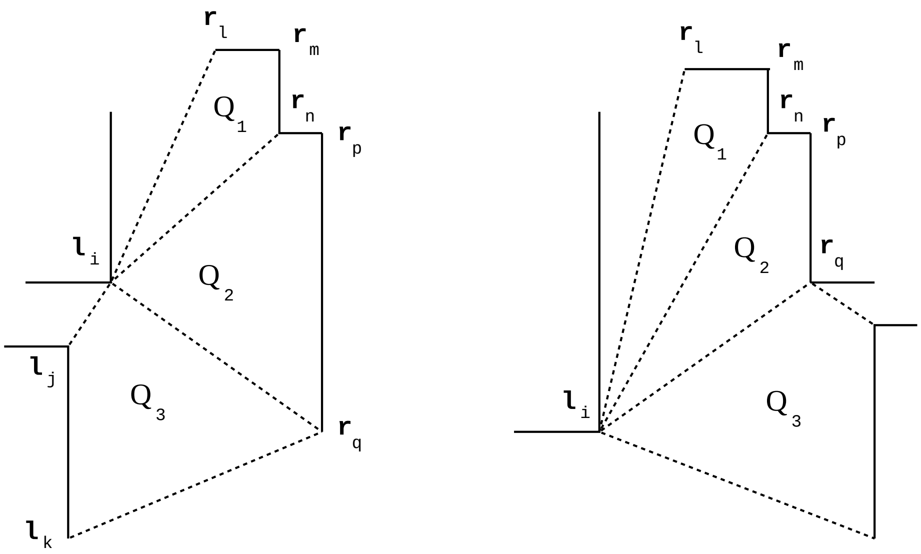


Figure 25: All three cross-quadrilaterals are of type B, the top two have single vertices on opposite sides.

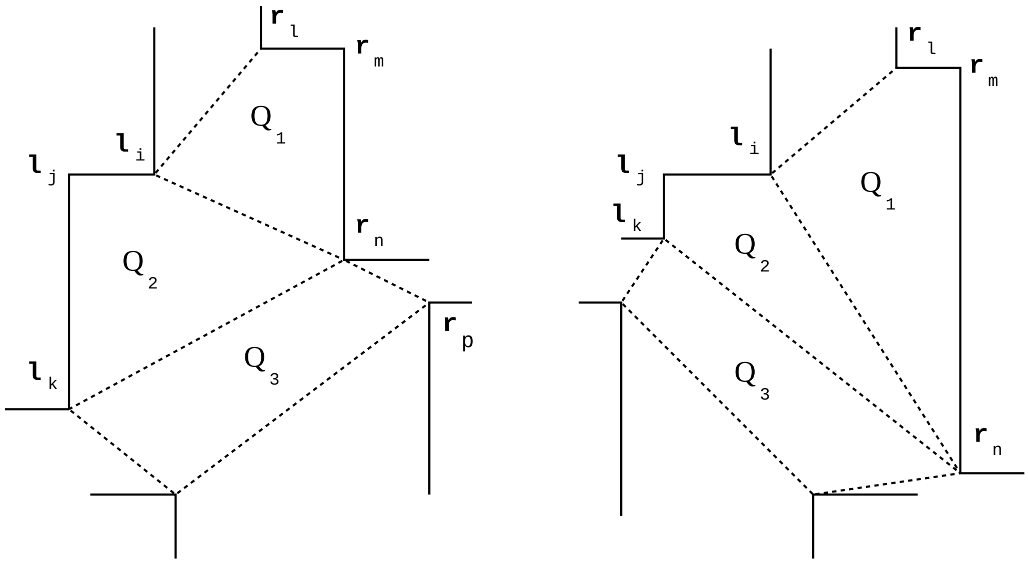


Figure 26: The top two cross-quadrilaterals are of type B, the bottom of type D.

will be on the left side and the bottom edge on the right side. Side-quadrilaterals on the left side are spanned by edge $r_m r_n$ and are seen by a guard on this edge. This guard also sees the three cross-quadrilaterals. Any side-quadrilaterals on the right side are left for the next guard assignment. Next consider the case that l_i lies below r_n (see figure 27). Q_2 has

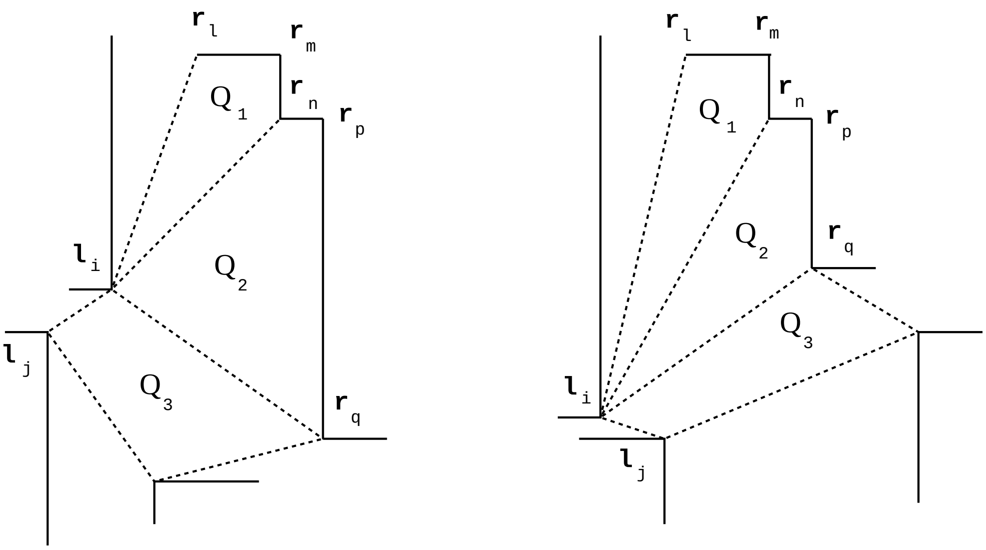


Figure 27: The two top cross-quadrilaterals are of type B, the bottom of type D.

vertex l_i on the left side and vertices $r_n r_p$ and r_q on the right side. If l_i lies above r_q then Q_3 has a top edge $l_i l_j$ on the left side and a bottom edge on the right side. A guard on edge $l_{i-1} l_i$ sees one side-quadrilateral between l_i and l_j and all three cross-quadrilaterals. Any side-quadrilaterals below r_q are left for the next guard assignment. If more than one side-quadrilateral lies between l_i and l_j then the new group consists of one such side-quadrilateral, Q_1 and Q_2. All other side-quadrilaterals and Q_3 are left for the next group. If instead l_i lies below r_q then $l_i l_j$ is a bottom edge of Q_3. A guard on edge $l_{i-1} l_i$ sees all side-quadrilaterals on the right side and the three cross-quadrilaterals. Side-quadrilaterals between l_i and l_j are left for the next guard assignment. $\diamond$

Case 2.3.7 *Exactly three cross-quadrilaterals are present. They are of types B, D and C.*

As explained in case 2.2.4, the top edge of Q_2 will lie on the left side or the right side depending on whether l_i lies above or below r_n (see figure 28). In both cases the bottom edge of Q_3 may lie on either the left or the right side of P. A guard on one of Q_2's horizontal edges always sees all three cross-quadrilaterals. Any present side-quadrilaterals are left for the next guard assignment. $\diamond$

Case 2.3.8 *Exactly three cross-quadrilaterals are present. They are of types B, D and A.*

No side-quadrilaterals are present in this configuration. None are present above Q_3 since the count at this point must be less than three and none are present next to Q_3 since Q_3 is

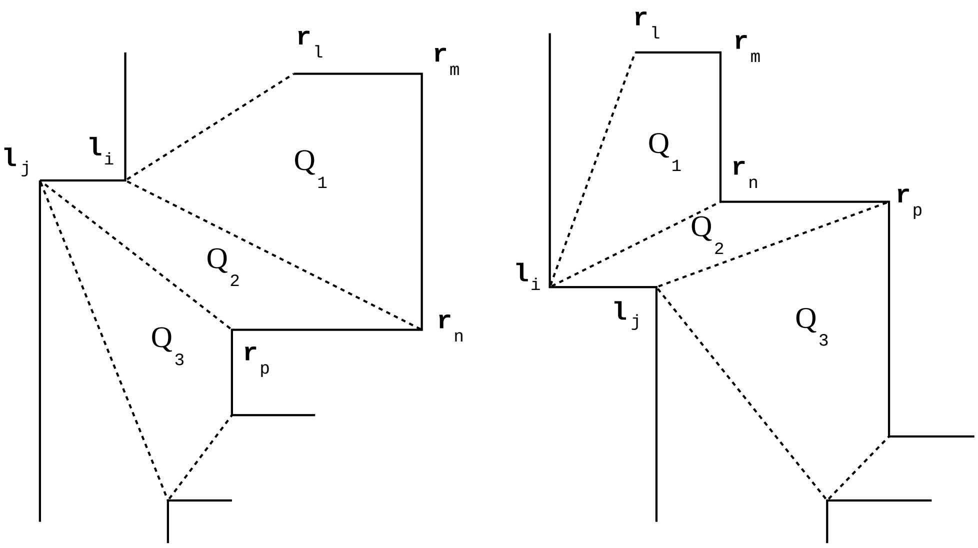


Figure 28: The top cross-quadrilateral is of type B, the middle of type D and the bottom of type C.

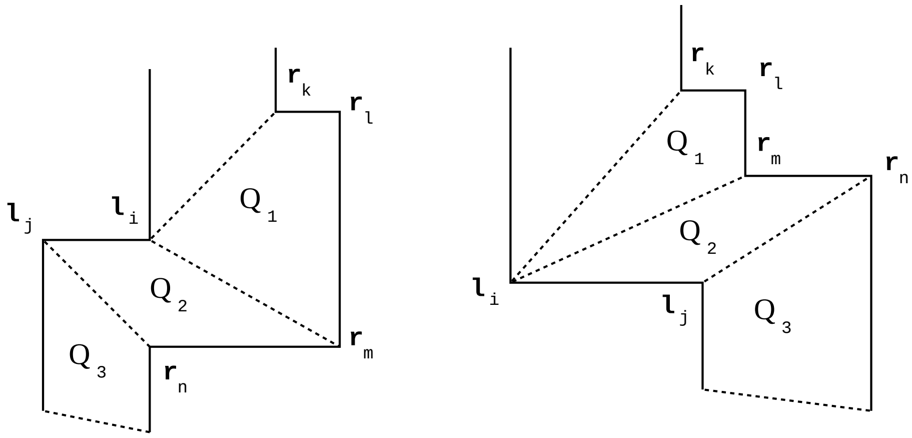


Figure 29: The top cross-quadrilateral is of type B, the middle of type D and the bottom of type A.

a type A quadrilateral with one vertical edge on each side of P (see figure 29). A guard on one of Q_2's horizontal edges sees all three cross-quadrilaterals. ◇

Next let the top cross-quadrilateral be of type C. Q_2 must then be of type C or type A. If Q_2 is of type C then Q_3 is of type C or A and if Q_2 is of type A then Q_3 is of type B or D. The possible combinations of types of three adjacent cross-quadrilaterals where the top is of type C are thus CCC, CCA, CAB and CAD.

Case 2.3.9 *Exactly three cross-quadrilaterals are present. All are of type C.*

As explained in case 2.2.5, Q_1 and Q_2 may have a single vertex on the same side of the P (see figure 30) or they may have single vertices on opposite sides of P (see figure 31). In both

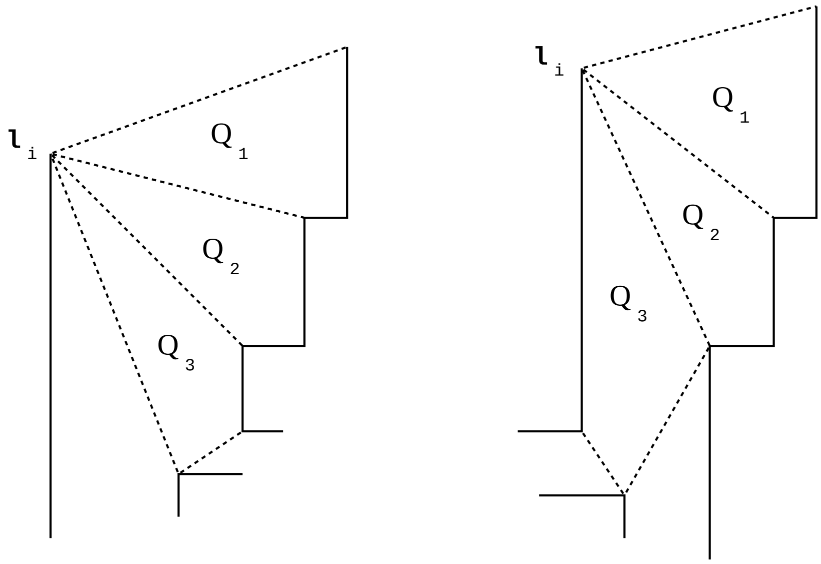


Figure 30: All cross-quadrilaterals are of type C. The top two have a single vertex on the left side.

cases Q_3 may have a single vertex on either side of P. Any side-quadrilaterals lie behind the bottom edge of Q_3. Assume first that both Q_1 and Q_2 have single vertex l_i on the left side of P. A guard on edge l_il_{i+1} sees all three cross-quadrilaterals. If the bottom edge of Q_3 lies on the right side, this guard also sees any present side-quadrilaterals. If the bottom edge of Q_3 lies on the left side then side-quadrilaterals are left for the next guard assignment. Assume next that Q_1 has a single vertex l_i on the left side of P and Q_2 has a single vertex r_j on the right side of P. A guard on edge r_jr_{j+1} sees all three cross-quadrilaterals. If the bottom edge of Q_3 lies on the left side, this guard also sees any present side-quadrilaterals. If the bottom edge lies on the right side of P then the side-quadrilaterals are left for the next guard assignment. ◇

Case 2.3.10 *Exactly three cross-quadrilaterals are present. They are of types C, C and A.*

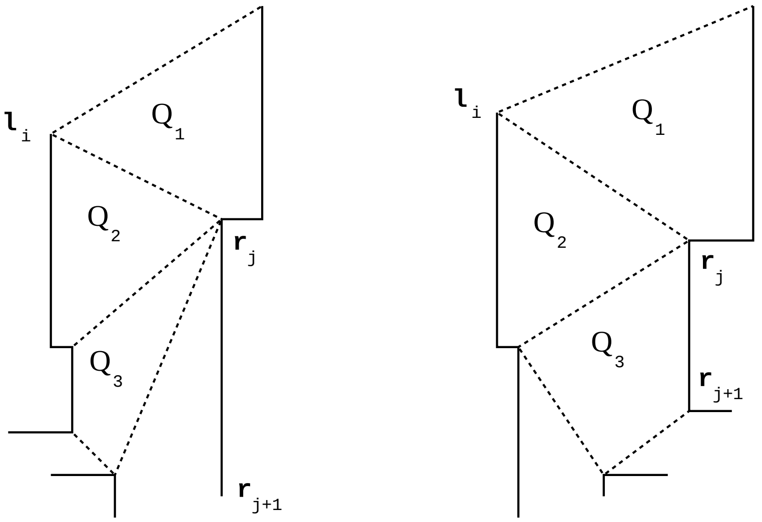


Figure 31: All cross-quadrilaterals are of type C. The top two have single vertices on opposite sides of P.

The only difference between this case and the previous case is that Q_3 is of type A with two vertical edges, one on each side of P (see figure 32). Thus no side-quadrilaterals are

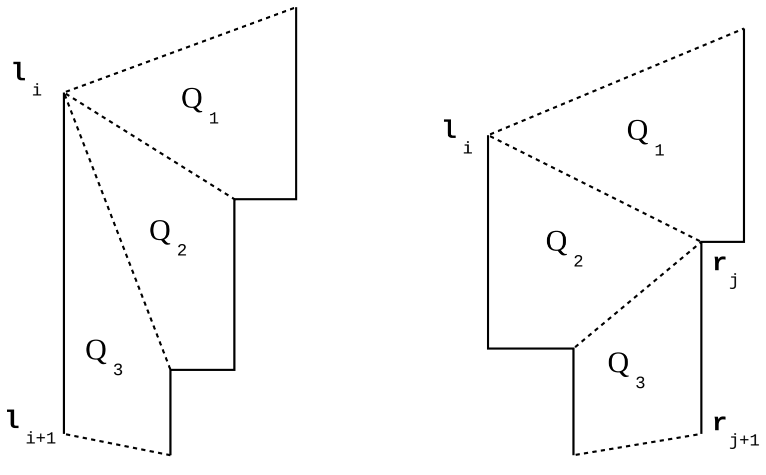


Figure 32: The top two cross-quadrilaterals are of type C, the bottom is of type A.

present and if Q_2's single vertex lies on the left side then a guard on edge l_il_{i+1} sees all quadrilaterals. If Q_2's single vertex lies on the right side then a guard on edge r_jr_{j+1} sees all quadrilaterals. $\diamond$

Case 2.3.11 *Exactly three cross-quadrilaterals are present. They are of types C, A and B.*

Assume that Q_1 has a single vertex l_i on the left side of P and let Q_2 have vertical edge l_il_j on the left side and r_kr_l on the right side of P (see figure 33). If l_j lies above r_l then

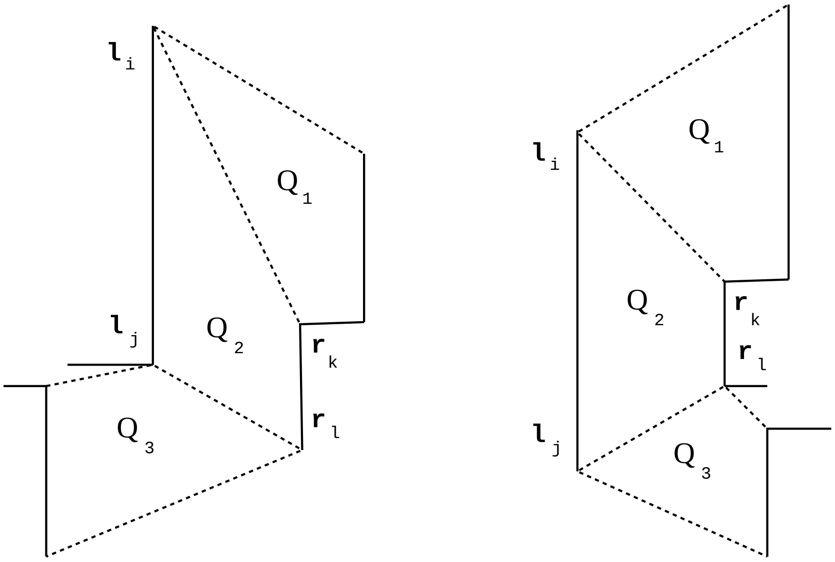


Figure 33: The top cross-quadrilateral is of type C, the middle of type A and the bottom of type B.

Q_3 will have a single vertex r_l on the right side and its remaining three vertices on the left side. Any side-quadrilaterals lie on the left side and are spanned by edge $r_k r_l$ A guard on this edge sees all quadrilaterals. If l_j lies below r_l then Q_3 has a single vertex l_j on the left side and three vertices on the right side. any side-quadrilaterals lie on the right side and are spanned by edge $l_i l_j$. A guard on this edge sees all quadrilaterals. $\diamond$

Case 2.3.12 *Exactly three cross-quadrilaterals are present. They are of types C, A and D.*

This case is similar to the previous case except that Q_3 is of type D and thus has a top edge on one side of P and a bottom edge on the opposite side of P (see figure 34). If l_j lies above r_l then Q_3's top edge lies on the left side and any side-quadrilaterals behind this edge are spanned by $r_k r_l$. A guard on this edge sees the side-quadrilaterals behind the top edge and the three cross-quadrilaterals. If l_j lies below r_l then Q_3's top edge lies on the right side and any side-quadrilaterals behind this edge are spanned by edge $l_i l_j$. A guard on this edge sees the side-quadrilaterals behind the top edge and the three cross-quadrilaterals. In both cases, any side-quadrilaterals behind the bottom edge of Q_3 are left for the next guard assignment. $\diamond$

Finally we consider the possible combinations of types when the top quadrilateral is of type D. With Q_1 of type D, Q_2 must be of type C or A. If Q_2 is of type C then Q_3 can be of type C or A and if Q_2 is of type A then Q_3 is of type B or type D. The possible combinations of types are thus DCC, DCA, DAB and DAD.

Case 2.3.13 *Exactly three cross-quadrilaterals are present. They are of types D, C and C.*

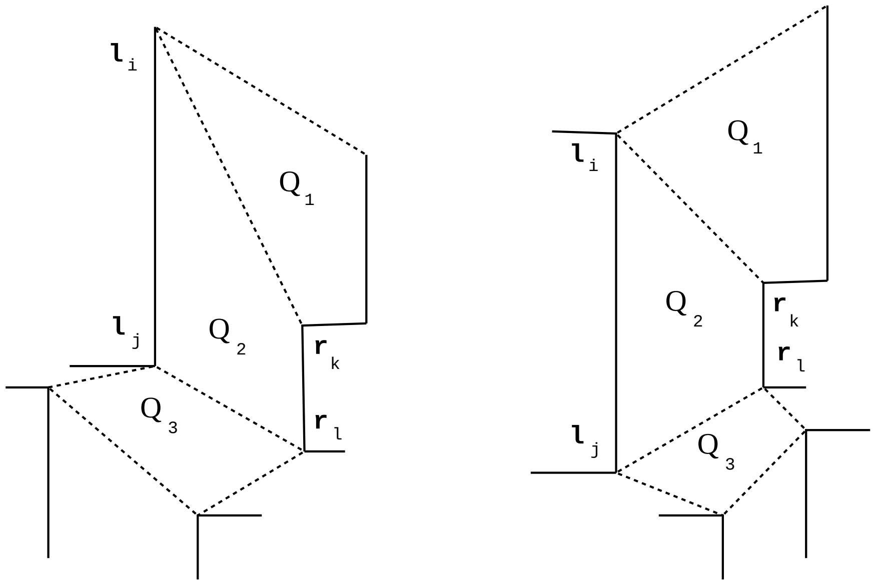


Figure 34: The top cross-quadrilateral is of type C, the middle of type A and the bottom of type D.

Assume that Q_1 has bottom edge $l_i l_j$ on the left side and top edge $r_k r_l$ on the right side of P. As explained in case 2.2.8, Q_2 may have its single vertex on either the left or the right side. The same is true for Q_3. Any side-quadrilaterals present lie behind the bottom edge of Q_3. First consider the case that Q_2 has single vertex l_i on the left side (see figure 35). If Q_3 also has single vertex l_i on the left side then edge $l_j l_{j+1}$ spans any present side-quadrilaterals. A guard on $l_j l_{j+1}$ sees all quadrilaterals. If instead Q_3 has a single vertex on the right side then the guard on edge $l_j l_{j+1}$ sees the cross-quadrilaterals. Side-quadrilaterals are left for the next guard assignment. Next consider the case when Q_2 has single vertex r_l on the right side (see figure 36). If Q_3 also has a single vertex on the right side then a guard on edge $r_l r_{l+1}$ sees all quadrilaterals. If Q_3 has a single vertex on the left side then the same guard sees the cross-quadrilaterals and any side-quadrilaterals are left for the next guard assignment. ◇

Case 2.3.14 *Exactly three cross-quadrilaterals are present. They are of types D, C and A.*

The difference between this case and the previous case is that Q_3 is of type A and no side-quadrilaterals can be present. This is so since a quadrilateral of type A has one vertical edge on each side of P (see figure 37). Thus with the same choice of guard as in case 2.3.13, all quadrilaterals are seen. ◇

Case 2.3.15 *Exactly three cross-quadrilaterals are present. They are of types D, A and B.*

Assume that Q_1 has bottom edge $l_i l_j$ on the left side and top edge $r_l r_m$ on the right side of P (see figure 38). The vertices of Q_2 are l_j and l_k on the left side and r_m and r_n on the right side. If l_k lies above r_n then Q_3 has a single vertex r_n on the right side and a top

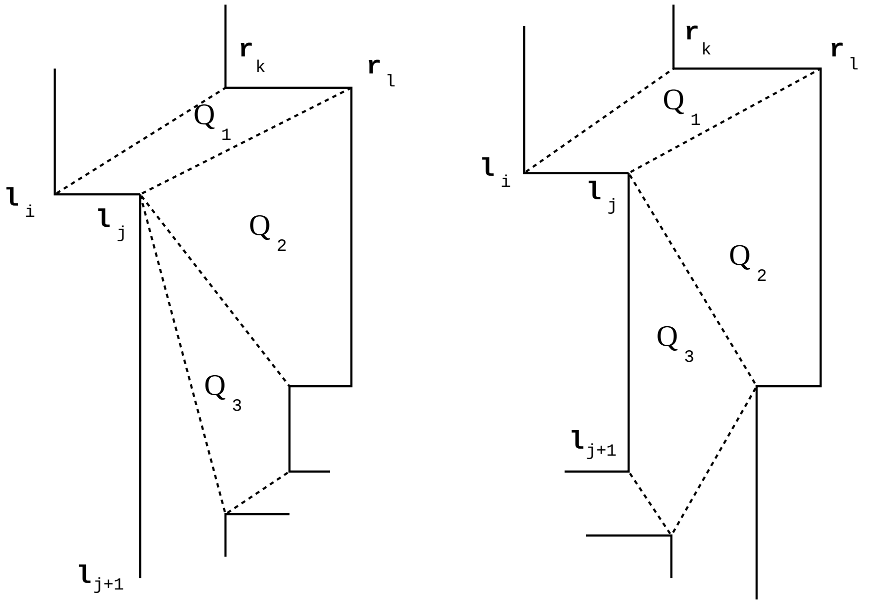


Figure 35: The top cross-quadrilateral is of type D, the other two of type C.

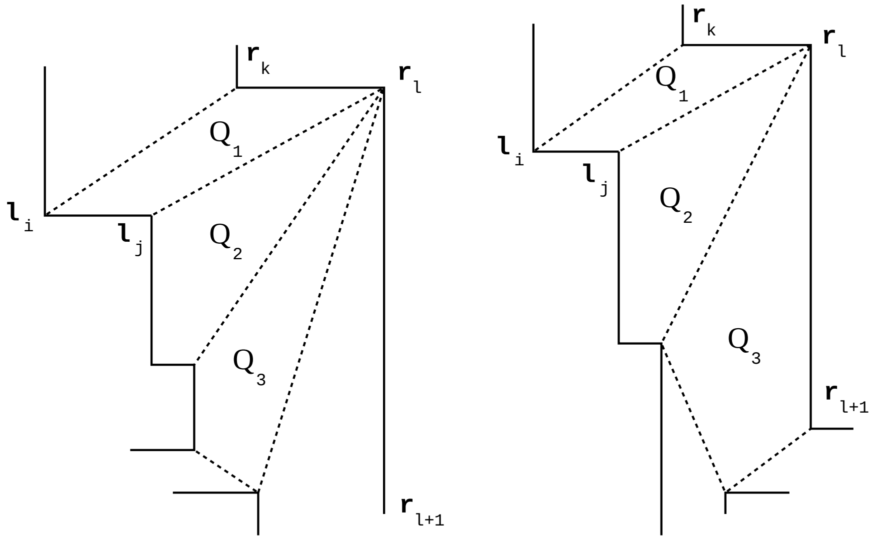


Figure 36: The top cross-quadrilateral is of type D, the other two of type C.

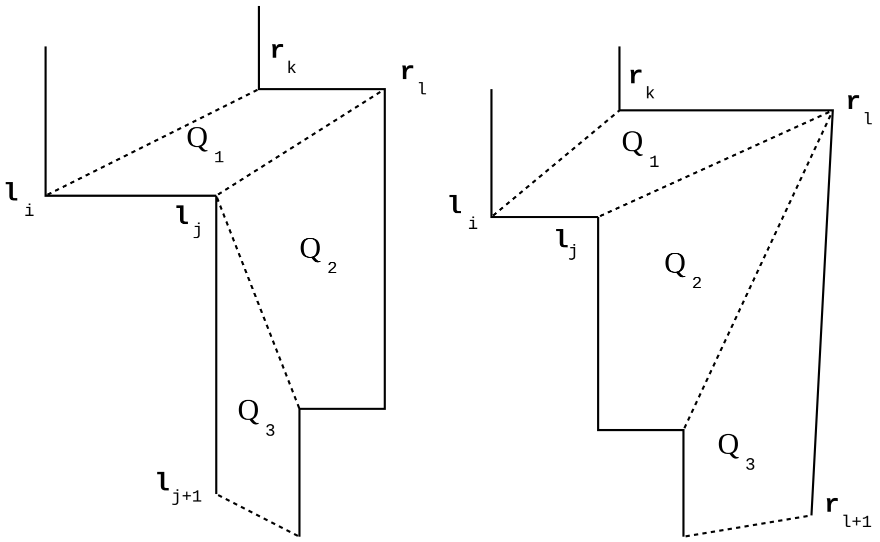


Figure 37: The top cross-quadrilateral is of type D, the middle of type C and the bottom of type A.

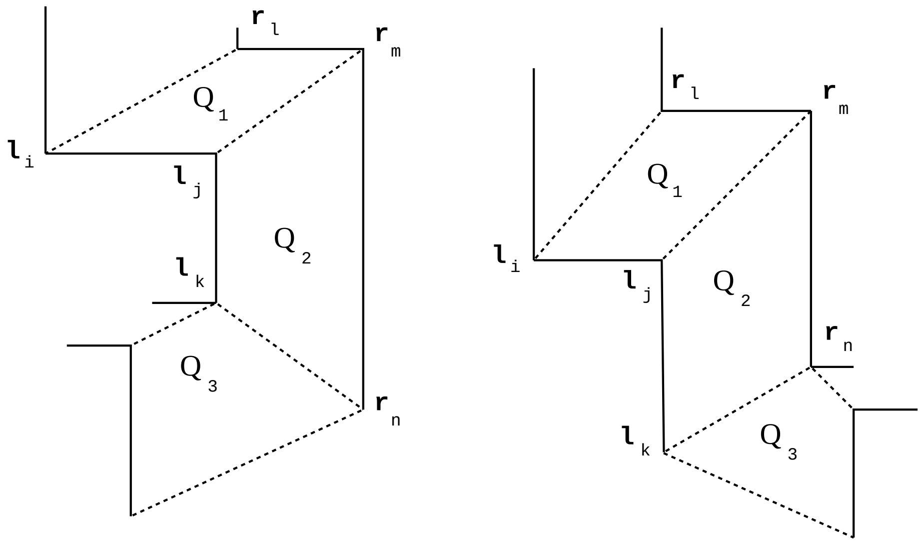


Figure 38: The top cross-quadrilateral is of type D, the middle of type A and the bottom of type B.

edge followed by a bottom edge on the left side. Side-quadrilaterals may lie behind the top edge. Any such side-quadrilaterals are spanned by edge $r_m r_n$ and a guard on this edge sees all quadrilaterals in the configuration. If instead l_k lies below r_n then Q_3 is in the reverse position and a guard on edge $l_j l_k$ sees all quadrilaterals. ◇

Case 2.3.16 *Exactly three cross-quadrilaterals are present. They are of types D, A and D.*

Let Q_1 and Q_2 be as described in the previous case (see figure 39). If l_k lies above r_n then

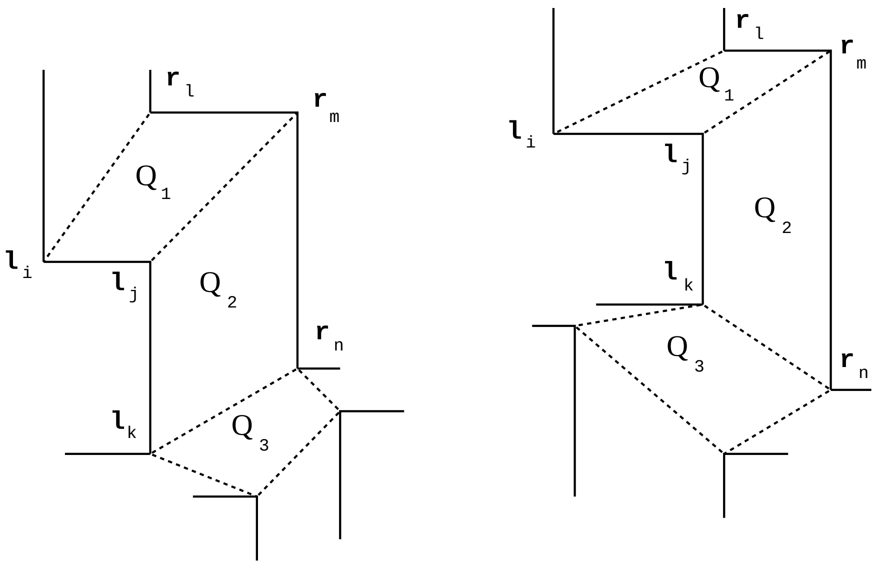


Figure 39: The top cross-quadrilateral is of type D, the middle of type A and the bottom of type D.

Q_3 has a top edge on the left side and a bottom edge on the right side. A guard on edge $r_m r_n$ sees any side-quadrilaterals behind the top edge and the three cross-quadrilaterals. If l_k lies below r_n then the top edge of Q_3 lies on the right side and a guard on edge $l_j l_k$ sees any side-quadrilaterals behind this edge and the three cross-quadrilaterals. In both cases any side-quadrilaterals behind the bottom edge are left for the next guard assignment. ◇

The above cases and their symmetric counterparts exhaust all possible configurations in the category where exactly three cross-quadrilaterals are present. Together with the previous two categories they ensure that an edge guard can always be chosen in such a way that all quadrilaterals in each group of at least three quadrilaterals are seen. The $\lceil (n-2)/6 \rceil$ edge guards assigned by the algorithm thus cover the whole polygon. Given the lower bound previously shown, we have the following theorem:

Theorem 1 $\lceil (n-2)/6 \rceil$ *edge guards are sometimes necessary and always sufficient to cover a monotone rectilinear polygon with n vertices.*

3 Algorithmic aspects

The upper bound in the previous section was obtained by taking an existing algorithm for the quadrilateralization of a polygon and modifying it to assign $\lceil (n-2)/6 \rceil$ edge guards in such a way that the polygon is covered.

The main modification to the quadrilateralization algorithm is the addition of a conditional statement, where each case corresponds to a particular configuration of quadrilaterals and results in the output of a guard. A count is kept on the number of unguarded quadrilaterals above the most recently generated cross-diagonal and the conditional statement is run through each time the count is at least three and also one last time when the last edge of the polygon has been processed.

To distinguish between cases we need to know the number and types of the cross-quadrilaterals, the number and positions of any side-quadrilaterals and the endpoints of the generated cross-diagonals. Therefore when the original algorithm outputs a cross-diagonal, we save this diagonal in an array CD. The type of the just created cross-quadrilateral is saved in an array $Type$. When the original algorithm outputs a side-diagonal, we increment a count of the number of side-quadrilaterals below the top endpoint of the just generated side-diagonal and we also increment the count of the number of side-quadrilaterals above the bottom endpoint of the same side-diagonal. These counts, called $Below(v)$ and $Above(v)$, are maintained for each vertex v of P. Each time a cross-diagonal is generated the $Above$ counts for the endpoints of the diagonal are included in the count of unguarded quadrilaterals. If the newly created cross-quadrilateral is of type B then the diagonal's endpoints are bottom vertices of vertical edges. In this case the $Above$ count for the top vertex of the quadrilateral's vertical edge is included. Since no configuration contains more than three cross-quadrilaterals, the size of array $Type$ is three and the size of CD is four.

The first step of the algorithm is to sort the horizontal edges of P in order of descending y-coordinate. Let the edges in sorted order be $e_1, ..., e_{n/2}$ and let the left and right endpoints of edge e_i be l_i and r_i, respectively. Also let the left and right endpoints of a diagonal d be l_d and r_d, respectively. As explained in section 2, the algorithm maintains a stack of top edges. Let the stack entries be numbered from 0 to top. With this notation the modified algorithm for the partitioning of a pyramid with bottom edge b is as follows:

Partition-pyramid(b):

$p \leftarrow 1$
$q \leftarrow 2$
while $q \leq$ top **do**
    Let v_p and v_q be the inner vertices of the pth and qth entries on the stack.
    if v_p and v_q lie on opposite chains **then**
        $d \leftarrow v_p v_q$
        $j \leftarrow j + 1$
        if $p = 1$ **then** $Type[j] \leftarrow$ A **else** $Type[j] \leftarrow$ B
        $CD[j] \leftarrow l_d r_d$
        if $Type[j] =$ A **then**
            $count \leftarrow count + 1$
        else
            $count \leftarrow count + 1 + Above(pred(v_x))$ where v_x is endpoint of d only
        if $count \geq 3$ **then** assign-a-guard
        $p \leftarrow p + 1$
    else $q \leftarrow q + 1$
    endif
endwhile
while $p \leq$ top **do**
    $d \leftarrow v_p v$ where v is endpoint of b on opposite chain from v_p
    $j \leftarrow j + 1$
    if $p = 1$ **then** $Type[j] \leftarrow$ A **else** $Type[j] \leftarrow$ B
    $CD[j] \leftarrow l_d r_d$
    if $Type[j] =$ A **then**
        $count \leftarrow count + 1$
    else
        $count \leftarrow count + 1 + Above(pred(v_x))$
    if $count \geq 3$ **then** assign-a-guard
    $p \leftarrow p + 1$
endwhile
$j \leftarrow j + 1$
if $top = 0$ **then** $Type[j] \leftarrow$ A **else** $Type[j] \leftarrow$ B
$CD[j] \leftarrow l_b r_b$
if $Type[j] =$ A **then**
    $count \leftarrow count + 1$
else
    $count \leftarrow count + 1 + Above(pred(l_x))$
    if $count \geq 3$ **then** assign-a-guard
    empty stack
    end

The main algorithm in its modified form becomes:

Algorithm:
Sort the edges
Push(e_1)
$i \leftarrow 2$
$CD[0] \leftarrow l_1 r_1$
$j \leftarrow 1$
while $i < n/2$ **do**
    if e_i is a top edge **then**
        push(e_i)
        $i \leftarrow i + 1$
    else (e_i is a bottom edge)
        $s \leftarrow$ top(stack)
        if e_i and s lie on opposite chains **then**
            $d_1 \leftarrow l_i l_s$; $d_2 \leftarrow r_i r_s$
            $dh \leftarrow$ the higher of the two diagonals d_1 and d_2
            $dl \leftarrow$ the lower of the two diagonals d_1 and d_2
            partition-pyramid(dh)
            push(dl)
            $j \leftarrow j + 1$
            $Type[j] \leftarrow$ D
            $CD[j] \leftarrow l_{dl} r_{dl}$
            $count \leftarrow count + 1 + Above(l_{dl}) + Above(r_{dl})$
            if $count \geq 3$ **then** assign-a-guard
            $i \leftarrow i + 1$
        else (e_i and s have at least one endpoint each on the same chain)
            $d \leftarrow v_i v_s$ where v_i and v_s are inner endpoints of e_i and s
            if d is a side-diagonal **then**
                $Above(v_i) \leftarrow Below(v_s) \leftarrow max(Above(v_i), Below(v_s)) + 1$
            else
                $j \leftarrow j + 1$
                $Type[j] \leftarrow$ A
                $CD[j] \leftarrow l_d r_d$
                $count \leftarrow count + 1 + Above(v_i)$
                if $count \geq 3$ **then** assign-a-guard
            endif
            $e_i \leftarrow d$
        endif
    endif
endwhile
partition-pyramid($e_{n/2}$)
if $count > 0$ **then** assign-a-guard
end

Assign-a-guard is a conditional statement that distinguishes among the three categories of cases. When this statement is executed, index j of arrays Type and CD is the same as the current number of unguarded cross-quadrilaterals.

assign-a-guard
if $j = 1$ **then** category1
elseif $j = 2$ **then** category2
else category3
end

Within each category, we distinguish between cases by checking the types of the present cross-quadrilaterals, deciding which of two diagonal endpoints has the larger y-coordinate and possibly checking where side-quadrilaterals are present. The variable $oldSQ$ keeps track of the number of side-triangles that may have been left unguarded during the previous guard assignment and $newSQ$ does the same for the current guard assignment. For the category where exactly one cross-quadrilateral is present we have:

category1:
$newSQ \leftarrow 0$
if $Type[1] = \text{A}$ **then**
    if $l_{CD[0]}$ lies below $r_{CD[0]}$ **then**
        output $r_{CD[0]} r_{CD[1]}$
    else (the case is symmetric)
        output $l_{CD[0]} l_{CD[1]}$
    endif
elseif $Type[1] = \text{B}$ **then**
    if $l_{CD[0]}$ lies below $r_{CD[0]}$ **then**
        output $pred(l_{CD[0]}) l_{CD[0]}$
    else (the case is symmetric)
        output $pred(r_{CD[0]}) r_{CD[0]}$
    endif
if $Type[1] = \text{C}$ **then**
    if $l_{CD[0]} = l_{CD[1]}$ **then** (the quadrilaterals has a single vertex on the left side)
        if $l_{CD[0]}$ lies below $r_{CD[0]}$ and $oldSQ \geq 2$ **then**
            output $r_{CD[0]} succ(r_{CD[0]})$
            $newSQ \leftarrow Above(r_{CD[1]})$
        else
            output $l_{CD[0]} l_{CD[1]}$
        endif
    else (the quadrilateral has a single vertex on the right side)
        the case is symmetric
    endif
if $Type[1] = \text{D}$ **then**

if $l_{CD[0]}$ lies below $r_{CD[0]}$ **then**
  if $Above(r_{CD[1]}) \geq 2$ **then**
    output $pred(l_{CD[0]})l_{CD[0]}$
    $newSQ \leftarrow Above(l_{CD[1]})$
  else
    output $r_{CD[1]}succ(r_{CD[1]})$
  endif
else (the case is symmetric)
endif
$oldSQ \leftarrow newSQ$
$count \leftarrow oldSQ$
$CD[0] \leftarrow CD[1]$
$j \leftarrow 0$ **end**

The routines for categories where exactly two and three cross-quadrilaterals are present are similar.

Each time a guard is assigned, the algorithm checks a constant number of cases before deciding which case is appropriate. These checks consist of looking up entries in the arrays CD and $Type$ and of comparing the y-coordinates of two known endpoints. All such checks take constant time and therefore it takes constant time to assign one guard once the quadrilaterals in a group have been generated. The original quadrilateralization algorithm generates the quadrilaterals in $O(n)$ time. Since each call to assign-a-guard results in the output of one guard and since $O(n)$ guards are assigned, the modified algorithm takes $O(n)$ time total.

4 Conclusions

In this paper we showed a tight bound for the number of edge guards required to cover a monotone rectilinear polygon. The upper bound was obtained by partitioning the polygon into convex quadrilaterals, collecting the quadrilaterals into groups of at least three and assigning one guard per group. The constructive proof produces a linear time algorithm for placing the guards. For rectilinear polygons that are not monotone, the problem of finding a tight bound on the number of required edge guards remains open. Collecting the quadrilaterals into groups of three and assigning one edge guard per group is no longer possible. Instead at least three edge guards may be necessary to cover a group of eight quadrilaterals. A polygon where this is the case is shown in figure 40.

5 Acknowledgemnts

The author wishes to thank Diane Souvaine for helpful comments and proofreading of the paper.

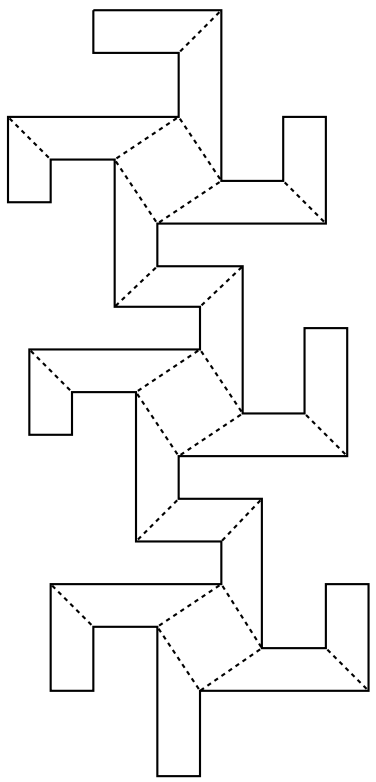


Figure 40: A non-monotone rectilinear polygon, which requires at least three edge guards per group of eight quadrilaterals.

References

[1] Aggarwal, A. "The Art Gallery Theorem: Its Variations, Applications and Algorithmic Aspects," Ph.D. thesis, Johns Hopkins University, 1984.

[2] Bjorling-Sachs, I. and Souvaine, D. "A Tight Bound for Edge Guards in Monotone Polygons," DIMACS Technical Report 92-52, December 1992.

[3] Chvatal, V. "A Combinatorial Theorem in Plane Geometry," *J. Combinatorial Theory Series* B 18 (1975), 39-41.

[4] Garey, Johnson, Preparata and Tarjan, "Triangulating a Simple Polygon," *Info. Proc. Letters* 7 (1978), 175-180.

[5] O'Rourke, J. "Art gallery problems and algorithms," Oxford University Press (1987).

[6] Sack, J.R. and Toussaint, G. "Guard Placement in Rectilinear polygons," *Computational Morphology* (1988), 153-175.

[7] Shermer, T. "Recent Results in Art Galleries," *Proceedings of the IEEE* vol. 80, no. 9, (1992), 1384-1399.