

# 2

## ORTHOGONAL POLYGONS

### 2.1. INTRODUCTION

In this chapter we consider *orthogonal polygons*, an important subclass of polygons that yield many interesting partitioning and art gallery theorems. An orthogonal polygon is one whose edges are all aligned with a pair of orthogonal coordinate axes, which we take to be horizontal and vertical without loss of generality.<sup>1</sup> Thus the edges alternate between horizontal and vertical, and always meet orthogonally, with internal angles of either  $90^\circ$  or  $270^\circ$ . Orthogonal polygons are useful as approximations to polygons; and they arise naturally in domains dominated by Cartesian coordinates, such as raster graphics, VLSI design, or architecture.

The orthogonal art gallery theorem was first formulated and proved by Kahn, Klawe, and Kleitman in 1980 (Kahn *et al.* 1983). It states that  $\lfloor n/4 \rfloor$  guards are occasionally necessary and always sufficient to see the interior of an orthogonal art gallery room. Thus the constrained nature of an orthogonal polygon permits covering with three-fourths as many guards as are needed for unrestricted polygons. Several different proofs of this theorem have been discovered, and several associated algorithms developed. As Fisk's proof of the unrestricted art gallery theorem eclipsed Chvátal's original proof, so Kahn *et al.*'s proof has been eclipsed by simpler proofs. But, as with Chvátal's proof, the original proof still retains considerable interest in its own right. So we will start with Kahn *et al.*'s proof, which establishes a beautiful partitioning result that is as important for orthogonal polygons as triangulation is for polygons: namely, that every orthogonal polygon may be partitioned by diagonals between vertices into convex quadrilaterals. The next section concentrates on establishing this theorem, from which the orthogonal art gallery theorem follows easily.

---

1. These polygons commonly have been called "rectilinear" polygons in the literature, but Grünbaum pointed out to me that "rectilinear" has the well-established meaning "characterized by straight lines," so that *every* polygon is rectilinear. Other terms used include "isothetic" and "rectanguloid."

## 2.2. KAHN, KLAWE, KLEITMAN PROOF

### 2.2.1. Convex Quadrilateralization

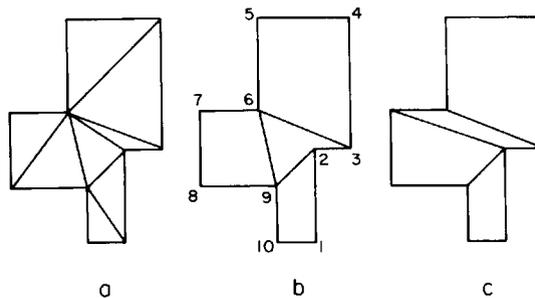
That a polygon can be partitioned with diagonals into triangles is almost obvious, perhaps because connecting any two vertices that can see one another is a valid first step in forming a triangulation: no care is required. Such is not the case with convex quadrilateralization: considerable care is required. Grouping pairs of triangles from a triangulation is *not* sufficient, as shown by Fig. 2.1a: no pairing of triangles in the illustrated triangulation leads to a convex quadrilateralization. I believe the main difference is the difficulty of finding “orthogonal ears.” This is illustrated in Fig. 2.1b. It is natural to consider the quadrilaterals (1, 2, 9, 10), (3, 4, 5, 6), and (6, 7, 8, 9) as “ears,” but removing them leaves the non-convex quadrilateral (2, 3, 6, 9). The unique convex quadrilateralization of this polygon is shown in Fig. 2.1c, which shows that (6, 7, 8, 9) is not an ear, although (1, 2, 9, 10) and (3, 4, 5, 6) are. In general, convex quadrilateralization is not unique, as demonstrated in Fig. 2.2.

Henceforth we will shorten “convex quadrilateralization” to “quadrilateralization”; the only quadrilateralizations that will be used in this book are convex quadrilateralizations.

The concept that plays the role of an “ear” is what Kahn *et al.* call a “tab.” To define this, we must first study the *neighbor* relation. Let  $P$  be an orthogonal polygon. Call a horizontal edge of  $P$  a *top* edge if the interior of  $P$  lies below it, and a *bottom* edge if the interior lies above it; *left* and *right* edges are defined similarly. A top edge  $T$  and a bottom edge  $B$  are *neighbors* if:

- (a)  $T$  and  $B$  can see one another (that is, there are points  $t$  and  $b$  on  $T$  and  $B$ , respectively, such that  $tb$  is never exterior to  $P$ ),
- (b) there is no bottom edge  $B'$  higher than  $B$  such that  $T$  can see  $B'$ , and
- (c) there is no top edge  $T'$  lower than  $T$  such that  $B$  see  $T'$ .

Note that the neighbor relation is symmetric by definition. Not every



**Fig. 2.1.** Neither triangle pairing (a) nor ear removal (b) can lead to the unique quadrilateralization (c).

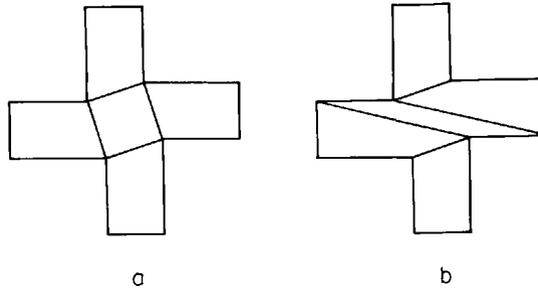


Fig. 2.2. Quadrilateralization is not unique.

horizontal edge has a neighbor, but if it does, it is clearly unique. So the relation matches certain pairs of top and bottom edges; see Fig. 2.3 for examples. We will see later that there must be at least one pair of neighboring edges in any orthogonal polygon.

A *tab* is a pair of neighboring edges connected to each other by a vertical edge. In Fig. 2.3a,  $(ab, cd)$  and  $(ef, gh)$  are tabs. What makes tabs important for convex quadrilateralization is that they can only be quadrilateralized in one way: in Fig. 2.3a, the quadrilaterals  $abcd$  and  $efgh$  must be part of any convex quadrilateralization. This will be proved in Lemma 2.3.

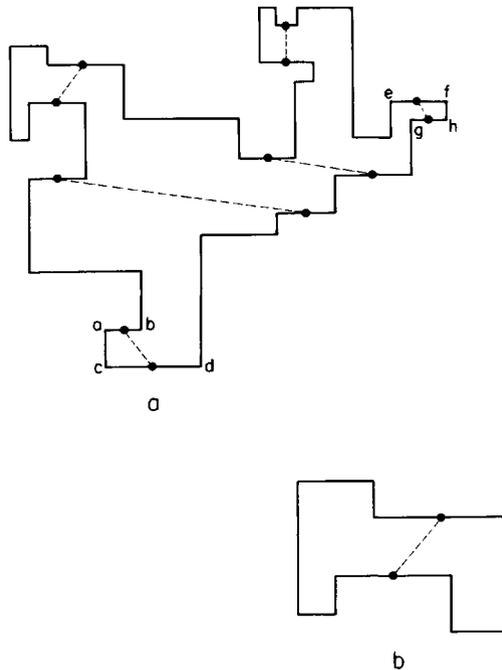


Fig. 2.3. Neighboring top and bottom edges;  $(ab, cd)$  and  $(ef, gh)$  are tabs.

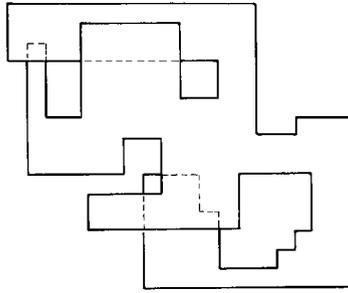


Fig. 2.4. An orthogonal polygon lying on several levels.

Unfortunately, it is not true that every orthogonal polygon has a tab; Fig. 2.3b shows an example that does not. Moreover, the concept can obviously be extended to define vertical tabs, but it is not even true that every orthogonal polygon must have either a horizontal or a vertical tab: Fig. 2.2 provides a counterexample. We will see below that Kahn *et al.* were forced to use a more complex structural characterization to achieve their result.

We now proceed with the proof. It is long and complicated. The proof is inductive, showing that any orthogonal polygon is reducible to a “smaller” one, which is convexly quadrilateralizable by the induction hypothesis. It is shown that any orthogonal polygon has at least one of three structural features:

- (1) neighboring edges that do not form a tab;
- (2) a “good” tab; or
- (3) a “tab pair.”

(These terms will be defined later.) The presence of these features is the “hook” that allows the reduction: for polygons without holes, the reduction amounts to cutting the polygon into two pieces, convexly quadrilateralizing each, then suturing the two quadrilateralizations together at the cut to form a convex quadrilateralization of the original.

The proof is remarkably general: it not only holds for orthogonal polygons, but also for orthogonal polygons with orthogonal holes, and also for orthogonal polygons that self-overlap in such a way that they can be considered to lie on several levels connected by “ramps.” Figure 2.4 shows an example. A precise technical definition of the class is: a orthogonal polygon on a Riemann surface corresponding to a function with singularities outside of itself. We note that the triangulation theorem could be similarly extended to the analogous class of unrestricted polygons.

### ***Geometric Lemmas***

The first lemma permits degeneracies to be ignored. Define an orthogonal polygon to have its vertices in *general position* if no two vertices have the

same horizontal or vertical coordinate. In the remainder of the section we will often shorten “orthogonal polygon” to “polygon” when there is no possibility of confusion.

**LEMMA 2.1.** A polygon  $P$  that is *not* in general position has the same quadrilateralization as any “nearby”  $P'$  that *is* in general position.

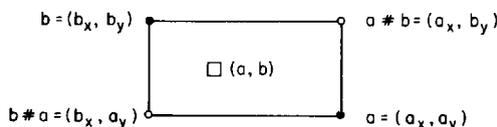
*Proof.*<sup>2</sup> Consider a sequence of orthogonal regions with the same number of edges as  $P$ , all in general position, that converge to  $P$ ; distance between regions is measured by the Hausdorff metric. Since there are only finitely many quadrilateralizations of these regions, each region in the “tail” of the sequence must have the same convex quadrilateralization. Since convexity is closed under taking limits, this quadrilateralization must also be a convex quadrilateralization for  $P$ .  $\square$

We will follow (Kahn *et al.* 1983) in first demonstrating the reductions, and then establishing the structural characterization that guarantees reducibility. We first need a geometric fact about neighboring edges. If  $a = (a_x, a_y)$  and  $b = (b_x, b_y)$  are two points, let  $a\#b$  be the point  $(a_x, b_y)$ , and let  $\square(a, b)$  be the rectangle or box with vertices  $a, b, a\#b, b\#a$ . Figure 2.5 illustrates these definitions. If  $H$  and  $V$  are a horizontal and vertical edge, respectively, then  $H\#V$  represents the point on the intersection of the lines containing  $H$  and  $V$ .

**LEMMA 2.2.** Let  $T$  and  $B$  be neighboring top and bottom edges of a polygon  $P$ . Then there is a left edge  $L$  left of both  $T$  and  $B$ , whose top endpoint is at least as high as  $T$  and whose bottom endpoint is at least as low as  $B$ , and a right edge  $R$  with analogous properties, such that  $\square(L\#B, R\#T)$  is completely interior to  $P$ .

*Proof.* Since  $T$  and  $B$  are neighbors, a point  $t$  on  $T$  sees a point  $b$  on  $B$ . We can clearly choose these to be interior points of  $T$  and  $B$ . For any point  $p$  interior to  $P$ , define the left-bounding edge to be the first vertical edge hit by a horizontal leftward ray from  $p$ . Choose  $L$  to be the rightmost of the left-bounding edges for the points of  $tb$  (we will see below that all the points of  $tb$  have the same left-bounding edge,  $L$ ).

Assume without loss of generality that  $t$  is left of  $b$  as illustrated in Fig. 2.6.  $L$  must be to the left of  $t$ , since otherwise, if  $L$  were between  $t$  and  $b$ ,  $T$



**Fig. 2.5.** Definition of the “ $\#$ ” and “ $\square$ ” symbols.

2. This proof assumes mathematical knowledge not used elsewhere in the book; it may be skipped without loss of continuity.

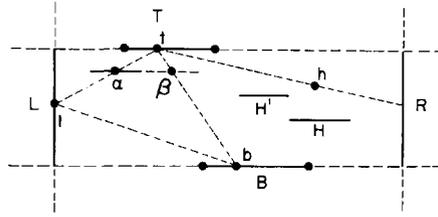


Fig. 2.6. If  $T$  and  $B$  are neighbors, then  $\square(L \# B, R \# T)$  is empty.

would see some bottom edge above the top of  $L$ , and so above  $B$ , contradicting the neighborliness of  $T$  and  $B$  (this claim is justified in more detail in Kahn *et al.* (1983)).

Let  $l$  be any point on  $L$  between  $T$  and  $B$ . Then  $l$  must be visible to both  $t$  and  $b$ . For suppose otherwise: then there must be a point  $\alpha$  of  $P$  on  $tl$  that blocks visibility. Let  $\beta$  be the point on  $tb$  horizontal from  $\alpha$ , as illustrated in Fig. 2.6. Then somewhere between  $\alpha$  and  $\beta$  there must be a vertical edge of  $P$ , which is the left-bounding edge for  $\beta$ , contradicting the fact that  $L$  is the rightmost left-bounding edge.

Therefore  $L$  must have its top at or above  $T$ , for otherwise  $T$  could see a bottom edge higher than  $B$ , contradicting the neighbor relation. Similarly,  $L$ 's bottom must be at or below  $B$ . Exactly analogous arguments establish the same properties for  $R$ , the leftmost right-bounding edge.

Finally, we show that  $Q = \square(L \# B, R \# T)$  is empty. Any vertical edge that intersects the interior of  $Q$  must have an endpoint in  $Q$ , for otherwise it would block the visibility of  $t$  for either  $L$  or  $R$ . So we can restrict discussion to horizontal edges. Let  $H$  be a horizontal edge that intersects  $Q$ . Draw a line of visibility from  $t$  to  $L$  or  $R$  (say  $R$ ) such that it passes above  $H$  at some point  $h$ , as illustrated in Fig. 2.6. Let  $H'$  be the horizontal edge of  $P$  that minimizes the vertical distance to a point on  $th$  [ $H'$  may be the same as  $H$ ]. Then  $H'$  is a bottom edge visible to  $T$  from  $t$ , contradicting the neighbor relation.  $\square$

This geometric fact implied by the neighbor relation leads to the crucial property of tabs.

**LEMMA 2.3.** If  $ab$  and  $cd$  are the horizontal edges of a tab, then any quadrilateralization must include the quadrilateral  $abcd$ .

*Proof.* Lemma 2.2 establishes that the situation is as illustrated in Fig. 2.7;

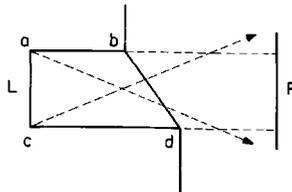


Fig. 2.7. A tab  $(ab, cd)$  forces the inclusion of quadrilateral  $abcd$ .

here  $L = ac$ . Any vertex visible to  $a$ , aside from  $b, c,$  and  $d$ , must lie below  $d$ . But connecting  $a$  to such a point means that  $c$  cannot be part of any convex quadrilateral. Similar arguments show that connecting  $c$  to any point above  $b$  blocks  $a$  from being part of a convex quadrilateral. Thus the quadrilateral  $abcd$  is necessary.  $\square$

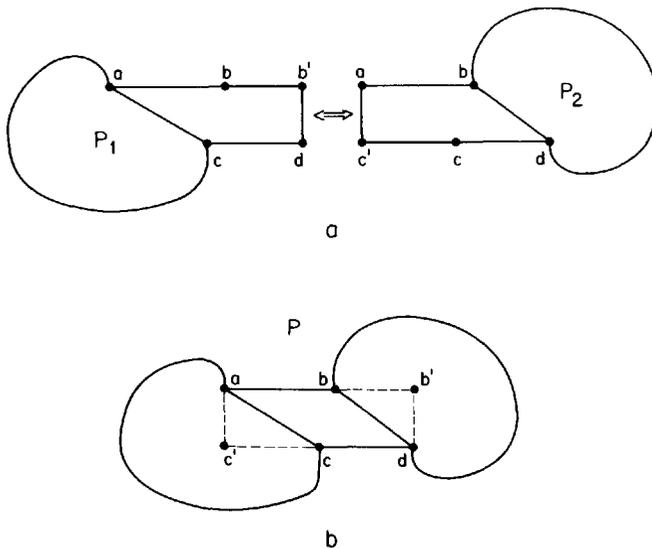
This is the key to the reductions: once a tab is isolated, the local quadrilateralization is known. We now proceed to describe the three reductions, after which the conditions supporting the reductions will be established.

**The Three Reductions**

We have yet to describe the quantity that is reduced by the reductions, and that forms the counter for the induction proof. A simple count of the number of vertices is not adequate because the reductions do not necessarily reduce the number of vertices. However, they either reduce the number of vertices or the number of holes. This suggests defining an orthogonal polygon  $P_1$  of  $h_1$  holes and  $n_1$  vertices as *smaller* than  $P_2$  with  $h_2$  and  $n_2$  holes and vertices if (1)  $h_1 < h_2$ , or (2)  $h_1 = h_2$  and  $n_1 < n_2$ . Thus, for polygons without holes, “smaller” just means fewer vertices. Finally, define a polygon to be *reducible* if, whenever every smaller polygon  $P'$  is quadrilateralizable, then so is  $P$ .

**LEMMA 2.4.** If  $P$  has a pair of neighboring edges that do not form a tab, then  $P$  is reducible.

*Proof.* Let the top edge  $T = ab$  and the neighboring bottom edge  $B = cd$  with  $a$  to the left of  $c$ . Let  $b' = d\#b$  and  $c' = a\#c$  as illustrated in Fig. 2.8b.



**Fig. 2.8.** Reduction for non-tab neighboring edges  $ab$  and  $cd$ .

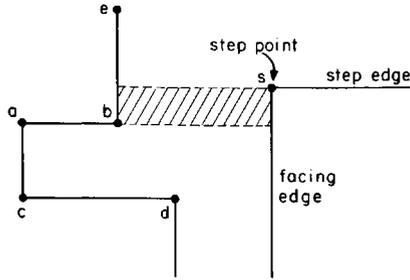
Note that the rectangle  $R = \square(a, d) = \square(c', b')$  is empty. Modify  $P$  to a multilevel polygon  $P'$  by introducing two tabs, one formed by the vertices  $a, b', d, c$ , and the other by the vertices  $d, c', a, b$ . In the special case when  $P$  has no holes,  $P'$  consists of two separate pieces  $P_1$  and  $P_2$ , as illustrated in Fig. 2.8a. In any case, if  $P'$  is disconnected, then both  $P_1$  and  $P_2$  are smaller than  $P$ , since each has no more holes than  $P$  but fewer vertices. We claim that if, on the other hand,  $P'$  is connected, then  $P'$  has fewer holes than  $P$ , and so is smaller. This claim may be established by the following argument.

Let  $\tau$  be a point immediately above  $T$  and  $\beta$  a point immediately below  $B$ . Both  $\tau$  and  $\beta$  are exterior to  $P$ , since  $T$  is a top edge and  $B$  a bottom edge. Now  $\tau$  and  $\beta$  are in the same connected component of the exterior of  $P'$ , as they may be connected by a path that skirts either of the new tabs. But  $\tau$  and  $\beta$  cannot be in the same connected component of the exterior of  $P$ : since  $P$  is connected, there must be a path within  $P$  that encircles either  $\tau$  or  $\beta$ , so that the cutting performed to make  $P'$  does not disconnect. Therefore, the reduction has reduced the number of holes of  $P$  by 1, and therefore  $P'$  is smaller.

Now assume the induction hypothesis: that all polygons smaller than  $P$  are quadrilateralizable; this guarantees that the reduced  $P'$  is quadrilateralizable. By Lemma 2.3, each of the introduced tabs can be quadrilateralized in just one way, as shown in Fig. 2.8a:  $ab'dc$  and  $bac'd$  must be included. Now, in  $P$ , replace these two quadrilaterals by  $abdc$  as shown in Fig. 2.8b, and otherwise use the remainder of the quadrilateralization of  $P'$ . The result is a quadrilateralization of  $P$ . We have shown therefore that  $P$  is reducible, establishing the lemma.  $\square$

The second reduction is based on the presence of certain types of tabs. This reduction is more complicated, and requires several definitions. Call a tab an *up tab* if its bottom edge extends horizontally further than its top edge, and a *down tab* if its top extends further than its bottom. Of the two bounding vertical edges guaranteed for a tab by Lemma 2.2, one connects the top to the bottom edge; call the other the *facing edge* of the tab. The top endpoint of the facing edge is called the *step point* and the adjacent horizontal edge the *step edge* for an up tab; for a down tab the step point and edge are at the bottom of the facing edge. These definitions are illustrated in Fig. 2.9.

Although tabs can be quadrilateralized in just one way, the mere presence of a tab does not lead immediately to a reduction. We classify tabs as either *good* or *bad*, depending on whether they do or do not lead to a reduction. Let  $ab$  and  $cd$  be the top and bottom edges of an up tab, and  $s$  its step point, as in Fig. 2.9. Then an up tab is *bad* if (1) its step edge is a bottom edge, and (2)  $\square(b, s)$  is empty. These conditions are illustrated in Fig. 2.9. A *good tab* is one that is not bad. Thus a good up tab is one either whose step edge is a top edge, or whose step edge is a bottom edge but there is an edge within  $\square(b, s)$ , and therefore necessarily a top edge. We will see below that the presence of a top edge in  $\square(b, s)$  permits the polygon to be cut near the tab in such a way as to establish reducibility.



**Fig. 2.9.** Definitions of tab components; the tab  $(ab, cd)$  is bad.

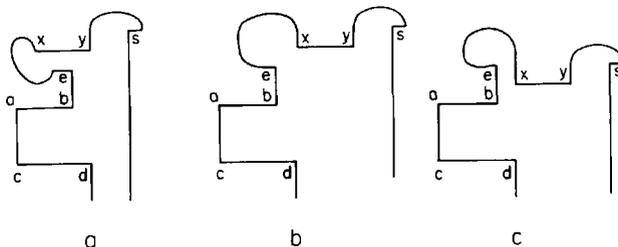
**LEMMA 2.5.** If  $P$  has a good tab, then  $P$  is reducible.

*Proof.* Assume that the tab is an up tab; the argument for a down tab can be obtained by turning every figure upside-down. Let the vertices be labeled as in Fig. 2.9:  $ab$  and  $cd$  are the top and bottom edges of the tab,  $e$  is adjacent to  $b$ , and  $s$  is the step point of the tab. Since the tab is good, either some edge intersects the interior of  $\square(b, s)$ , or the step edge is a top edge. In the former case let  $xy$  be the lowest edge that intersects  $\square(b, s)$ , and in the latter let  $xy$  be the step edge. In either case, let  $x$  be left of  $y$ .

The analysis proceeds with two cases:  $x$  is left of  $b$ , in which case  $x$  must also be above  $e$  (Fig. 2.10a), or  $x$  is right of  $b$ , in which case it may be above or below  $e$  (Figs. 2.10b and 2.10c). In all three figures, it may be that  $y = s$  so that the step edge is a top edge.

*Case 1* ( $x$  is left of  $b$  (Fig. 2.10a).). Replace  $xy$  and the chain  $e, b, a, c, d$  with two tabs, one down tab defined by the chain  $y, b \# y, b \# c, d$ , and one left horizontal tab defined by the chain  $x, y, y \# b, b, e$ . Call the modified polygon  $P'$ . If  $P$  has no holes, then these alterations separate  $P$  into two polygons  $P_1$  and  $P_2$ , as illustrated in Fig. 2.11a; otherwise the tabs overlap on different levels in  $P'$ .

Assume that any polygon smaller than  $P$  can be quadrilateralized. If  $P$  has no holes, then  $P'$  is clearly smaller, as both  $P_1$  and  $P_2$  have fewer vertices. If  $P$  has holes, then  $P'$  has one fewer hole. This can be seen by considering two exterior points  $\tau$  and  $\beta$ , with  $\tau$  above  $xy$  and  $\beta$  below  $cd$ ; the argument is identical to that used in Lemma 2.4. Thus in either case  $P'$  is smaller and can therefore be quadrilateralized. It will be easier to assume



**Fig. 2.10.** Three good tab cases.

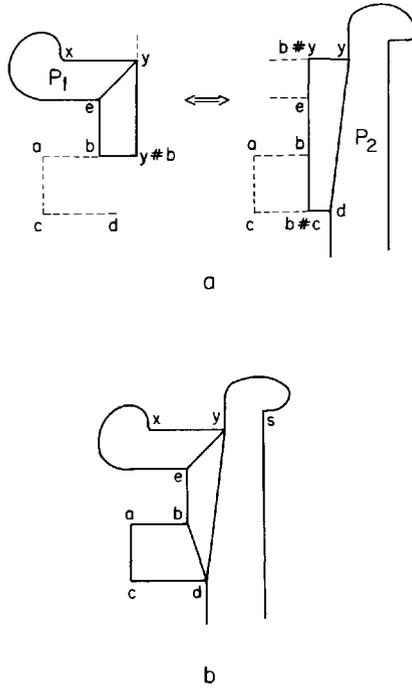


Fig. 2.11. Good tab reduction for Fig. 2.10a.

henceforth that  $P$  has no holes, although the argument is identical in the general case. By Lemma 2.3, the introduced tabs may only be quadrilateralized in one way: the quadrilateral  $(y, y\#b, b, e)$  is included in  $P_1$  and  $(y, b\#y, b\#c, d)$  is included in  $P_2$ , as illustrated in Fig. 2.11a. Note that the diagonal  $ey$  cuts off  $P_1$  and  $yd$  cuts off  $P_2$  in such a manner that the two quadrilateralizations can be put together as shown in Fig. 2.11b: the tab quadrilaterals are removed and replaced by  $yebd$ , and  $abcd$  is added. The result is a quadrilateralization of  $P$ , establishing that  $P$  is reducible.

*Case 2* ( $x$  is right of  $b$  (Figs. 2.10b and 2.10c).). The two situations illustrated in Figs. 2.10b and 2.10c are handled with the same reduction; we will use the case where  $x$  is above  $e$  (Fig. 2.10b) as illustrated. The replacements made are the same as in Case 1, but the argument is a bit different. Perform the same alterations as in Case 1, resulting, when  $P$  has no holes, in  $P_1$  and  $P_2$  as illustrated in Fig. 2.12a. As in Case 1,  $P'$  is smaller, and so can be quadrilateralized. The tab introduced to  $P_2$  requires the inclusion of the quadrilateral  $(y, b\#y, b\#c, d)$ , just as in Case 1, but  $(y, y\#b)$  and  $eb$  are no longer neighbors in  $P_1$ , and so do not form a tab. Nevertheless, we claim that either the diagonal  $ey$  or  $bx$  is a part of any quadrilateralization of  $P_1$ .

Suppose to the contrary that  $y\#b$  lies on more than one quadrilateral. Then a diagonal from  $y\#b$  must either (1) go to the left of  $eb$ , blocking  $b$

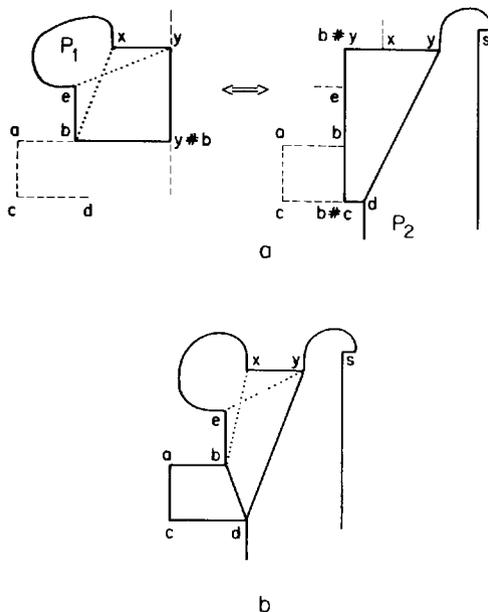


Fig. 2.12. Good tab reduction for Fig. 2.10b.

from being a vertex of any quadrilateral, or (2) go above  $xy$ , blocking any quadrilateral from containing  $y$ . Both (1) and (2) follow from the fact that  $\square(b, y)$  is empty, since  $xy$  was chosen to be the lowest edge that intersects  $\square(b, s)$ .

Thus either  $(e, y, y\#b, b)$  or  $(b, x, y, y\#b)$  is in the quadrilateralization of  $P_1$ . In the former case, we make the same replacements as in Case 1: replace the end quadrilaterals in  $P_1$  and  $P_2$  with  $yebd$ . In the latter case replace with  $ydbx$ . In both cases add  $abcd$ . The result is a quadrilateralization of  $P$ , illustrated in Fig. 2.12b, establishing that  $P$  is reducible.  $\square$

The third and final reduction depends on the presence of a tab pair: an up tab  $U$  and a down tab  $D$  such that the step edge of  $U$  is the bottom edge of  $D$  and the step edge of  $D$  is the top edge of  $U$ . Thus, as shown in Fig. 2.13b, the tabs “face” one another without intervening edges. Reduction is comparatively straightforward in this case.

**LEMMA 2.6.** If  $P$  contains a tab pair, then  $P$  is reducible.

*Proof.* Let  $ab, cd$  form the up tab, and  $fg, hi$  form the down tab, as illustrated in Fig. 2.13b. Move  $ab$  up to form the tab  $(a\#f, f)$ ,  $cd$ , and move  $fg$  down to form the tab  $(b, g\#b), hi$ . If  $P'$  is disconnected, then two pieces  $P_1$  and  $P_2$  (Fig. 2.13a) are both smaller than  $P$ . If  $P'$  is connected, then  $P'$  has one less hole than  $P$ , as can be seen by considering paths from an exterior point  $\lambda$  just left of  $ac$  and an exterior point  $\rho$  just right of  $ig$ . Thus

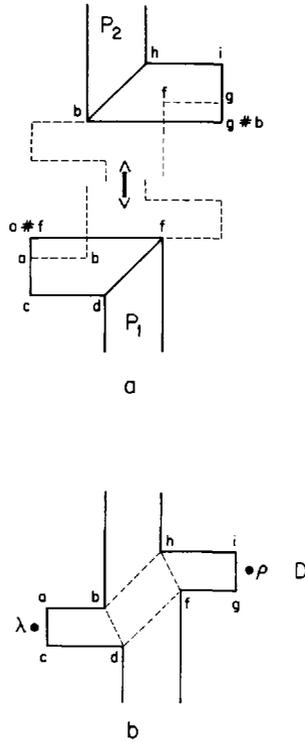


Fig. 2.13. Tab pair reduction.

$P'$  is smaller. The induction assumption then guarantees that each is quadrilateralizable, and Lemma 2.3 ensures the introduced tabs have a unique quadrilateralization, as shown in Fig. 2.13a. These can be replaced by  $abcd$ ,  $bdfh$ , and  $hifg$  in  $P$ , producing a quadrilateralization of  $P$  (Fig. 2.13b), and establishing that  $P$  is reducible.  $\square$

We now have established that if any one of the three structural features is present in  $P$ , then  $P$  is reducible—that is, the quadrilateralization theorem can be established by induction. We now turn to proving that in any orthogonal polygon, at least one of the three structures is present. The proof is by contradiction: we show that if  $P$  is irreducible, and therefore contains no instances of the three reducible structures, then  $P$  must have an infinite number of edges.

An important tool in the proof will be an association of every horizontal edge  $E$  with a tab  $tab(E)$  as follows. If  $E$  is a top edge, define  $n(E)$  to be the highest bottom edge below  $E$  that is visible to  $E$ ; if  $E$  is a bottom edge,  $n(E)$  is the lowest top edge above  $E$  visible to  $E$ . Clearly  $n(E)$  is well-defined for any  $E$ , so it may be composed an arbitrary number of times. Note that if  $n(n(E)) = n^2(E) = E$ , then  $E$  and  $n(E)$  are neighbors. Since  $P$  has a finite number of edges, the sequence  $E, n(E), n^2(E), \dots, n^k(E), n^{k+1}(E), n^{k+2}(E), \dots$  must be finite. Since visibility is

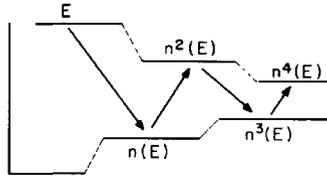


Fig. 2.14. The chain induced by  $n(E)$  leads to  $tab(E)$ .

symmetric it must be the case that  $n^{i+2}(E)$  falls between or at the heights of  $n^i(E)$  and  $n^{i+1}(E)$ : otherwise  $n^{i+1}(E)$  would map to  $n^i(E)$ , which would be closer and visible by symmetry. Therefore, the sequence cannot terminate with  $n^i(E) = n^j(E)$  with  $j - i > 2$ ; it must terminate with  $n^k(E) = n^{k+2}(E)$  for some  $k$ , as illustrated in Fig. 2.14. As we observed above, this implies that  $n^k(E)$  and  $n^{k+1}(E)$  are neighbors. Because we are assuming that  $P$  is irreducible, it cannot have any neighboring edges that do not form a tab, by Lemma 2.4. Therefore,  $n^k(E)$  and  $n^{k+1}(E)$  form a tab, which is designated as  $tab(E)$ . Moreover, since an irreducible polygon cannot contain a good tab by Lemma 2.5,  $tab(E)$  must be a bad tab.

We now prove that the relationship between  $E$  and  $tab(E)$  illustrated in Fig. 2.14 is the only possible configuration.

**LEMMA 2.7.** Let  $E$  be a horizontal edge,  $tab(E)$  its tab, and  $F$  the facing edge of  $tab(E)$ . Then if  $E$  is a top edge,  $F$  falls horizontally between  $F$  and the top edge of  $tab(E)$ , and if  $E$  is a bottom edge, between  $F$  and the bottom edge of  $tab(E)$ .

*Proof.* Without loss of generality let the tab  $tab(E)$  be an up tab with top edge  $n^{k+1}(E)$  and bottom edge  $n^k(E)$  as illustrated in Fig. 2.15, and consider the top edge  $n^{k-1}(E)$ . If it overlaps horizontally with  $n^{k+1}(E)$ , then it must see a bottom edge  $A$  that is higher than  $n^k(E)$ , and if it extends left of  $F$ , then it must see a bottom edge  $B$  that is higher than  $n^k(E)$ .<sup>3</sup> The same argument can be applied to  $n^{k-2}(E)$ , and so on, backwards to  $E$ .  $\square$

So far we have not explored the implications of the fact that since  $P$  is irreducible, each  $tab(E)$  must be a bad tab. The next lemma depends crucially on this constraint.

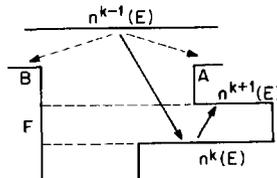


Fig. 2.15.  $E$  falls between  $F$  and the top edge of  $tab(E)$ .

3. See Kahn *et al.* (1983) for a detailed proof of these claims.

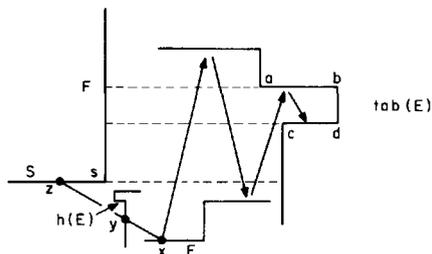


Fig. 2.16. There must exist a bottom edge  $h(E)$  in the illustrated situation.

**LEMMA 2.8.** Suppose  $P$  is irreducible and that  $E$  is a bottom edge such that  $tab(E)$  is a down tab not containing  $E$ . Then there is a bottom edge  $h(E)$  that is not part of a down tab.

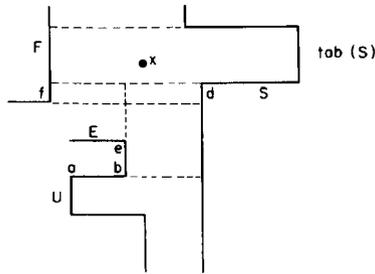
*Proof.* Let the top and bottom edges of the tab  $tab(E)$  be  $ab$  and  $cd$ , and let  $F$  be the facing edge,  $s$  the step point, and  $S$  the step edge, as illustrated in Fig. 2.16. Since  $tab(E)$  is a bad tab, (1)  $S$  is a top edge, and (2)  $\square(s, c)$  is empty. The second condition implies that  $E$  is below  $S$ . Now  $E$  cannot see  $S$ , since  $tab(E)$  must fall between  $E$  and  $n(E)$ . Thus a line  $xz$  from  $E$  to  $S$  must intersect an edge of  $P$ . Let  $y$  be the intersection closest to  $x$ . Clearly the edge through  $y$  cannot be a bottom edge, since  $x$  sees  $y$ , nor can it be a top edge, for the same reason that  $E$  cannot see  $S$ . So there must be a vertical edge through  $y$ . Let  $h(E)$  be the horizontal edge adjacent to the upper end point of this edge. If  $h(E)$  were a top edge, then either  $x$  sees it or one even lower, contradicting the fact that  $abcd$  is  $tab(E)$  and  $F$  faces this tab. Thus  $h(E)$  is a bottom edge. However,  $h(E)$  could not be the bottom edge of a down tab, for then  $x$  could either see the top of this tab, or some edge even lower.  $\square$

Application of this lemma repeatedly will show that irreducible polygons must have an infinite number of bottom edges. We must also employ the restriction that irreducible polygons cannot contain a tab pair.

**LEMMA 2.9.** If  $P$  is irreducible, then  $P$  has an infinite number of edges.

*Proof.* Suppose to the contrary that  $P$  has a finite number of edges. Then, as described previously, the sequence  $E, n(E), n^2(E), \dots$  must lead to a tab  $U$ , which we can assume without loss of generality to be an up tab. Let  $ab$  be the top edge of the highest up tab  $U$  of  $P$ , and let  $S$  be  $U$ 's step edge. We seek to establish that there is a bottom edge  $E$  that is above  $ab$  and not part of  $tab(E)$ ; then Lemma 2.8 will be applied to obtain a contradiction.

Because  $U$  is a bad tab, its step edge  $S$  must be a bottom edge. Now, since  $S$  is above  $ab$ ,  $tab(S)$  is above  $U$ , and since  $U$  is the highest up tab,  $tab(S)$  is a down tab. So if  $S$  is not part of  $tab(S)$ , then  $S$  can serve as  $E$ . So assume that  $S$  is the bottom edge of the down tab  $tab(S)$ . Let  $d$  be the step point of  $U$ ,  $e$  the upper endpoint of the vertical edge incident to  $b$ ,  $f$  the step point of  $tab(S)$ , and let  $E$  be the horizontal edge meeting  $e$ .  $U$  and  $tab(S)$



**Fig. 2.17.**  $U$  is the highest up tab, and  $S$  its step edge.

cannot form a tab pair (since  $P$  is irreducible), so  $f \neq b$ . Since  $U$  is a bad tab,  $\square(b, d)$  is empty, which implies that  $E$  is a bottom edge. Since  $\text{tab}(S)$  is also a bad tab,  $\square(d, f)$  is empty, so  $e$  must be lower than  $f$ . These relationships are illustrated in Fig. 2.17.

We now show that  $E$  is not part of  $\text{tab}(E)$ . Suppose to the contrary that  $E$  is the bottom edge of  $\text{tab}(E)$ , and let  $T$  be its top edge. Since  $\text{tab}(E)$  must be a down tab, there is a point  $x$  of  $T$  strictly between  $b$  and  $d$  horizontally.  $x$  must be above  $S$  since  $\square(b, d)$  is empty, and it must be below the top of  $\text{tab}(S)$ , since otherwise  $E$  and  $T$  would not be neighbors. But then  $x$  lies within the rectangle determined by  $\text{tab}(S)$ , guaranteed to be empty by Lemma 2.2. This contradiction establishes that  $E$  is not part of  $\text{tab}(E)$ .

Now we may apply Lemma 2.8 to  $E$  (since  $\text{tab}(E)$  is a down tab) to obtain a bottom edge  $h(E)$  above  $E$  that is not part of a down tab. Since again  $\text{tab}(h(E))$  must be a down tab,  $h(E)$  is not part of  $\text{tab}(h(E))$ , and the lemma is again applicable. Proceeding in this manner we obtain an infinite sequence of distinct edges  $E, h(E), h^2(E), \dots$ , thus establishing that  $P$  has an infinite number of edges.  $\square$

We can summarize the argument in the following theorem.

**THEOREM 2.1** [Kahn, Klawe, and Kleitman 1980]. Every orthogonal polygon  $P$  (with or without holes) is convexly quadrilateralizable.

*Proof.* By Lemma 2.1 it suffices to consider polygons whose vertices are in general position. The basis of the induction proof is established by a rectangle, which is itself a convex quadrilateral. Assume then that all polygons smaller than  $P$  can be quadrilateralized. Lemma 2.9 establishes that  $P$  must contain an instance of at least one of the following structures:

- (1) A pair of neighboring edges that do not form a tab.
- (2) A good tab.
- (3) A tab pair.

Lemmas 2.4, 2.5, and 2.6 show that each of these features permits  $P$  to be reduced to a smaller  $P'$  in such a way that a quadrilateralization for  $P'$  (available by the induction hypothesis) extends to a quadrilateralization for  $P$ .  $\square$

### 2.2.2. The Orthogonal Art Gallery Theorem

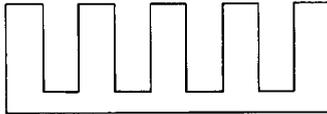
With Theorem 2.1 available, an easy proof of  $\lfloor n/4 \rfloor$  sufficiency for coverage of an orthogonal polygon without holes is possible along the same lines as Fisk's proof of  $\lfloor n/3 \rfloor$  sufficiency for general polygons.

**THEOREM 2.2** [Kahn, Klawe, and Kleitman 1980].  $\lfloor n/4 \rfloor$  guards are sometimes necessary and always sufficient to cover the interior of an orthogonal polygon of  $n$  vertices.

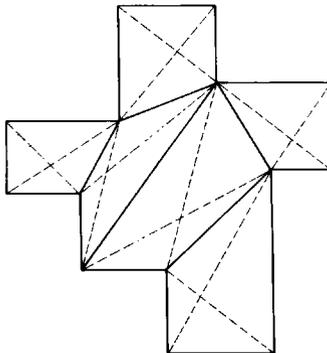
*Proof.* Necessity is established by the orthogonal version of Chvátal's comb example: one guard is needed for each tong in Fig. 2.18.

For sufficiency, construct a graph  $G$  from a quadrilateralization of  $P$  by adding both diagonals to each quadrilateral, as illustrated in Fig. 2.19. Although it is not immediately obvious,  $G$  is planar, and therefore 4-colorable. We can establish 4-colorability without invoking the Four Color Theorem as follows.

Let  $\tilde{Q}$  be the dual of the quadrilateralization of  $P$ : each node of  $\tilde{Q}$  corresponds to a quadrilateral, and two nodes are connected by an arc if their quadrilaterals share a side. Then  $\tilde{Q}$  must be a tree, for if it contained a cycle, this would imply that  $P$  has a hole. Now proceed by induction. Remove any leaf quadrilateral  $q$ , leaving the tree  $\tilde{Q}'$ . Since  $q$  has degree 1, it may be removed by cutting along a single diagonal  $d$  of the quadrilateralization. Four-color  $\tilde{Q}'$  by the induction hypothesis, and reattach  $q$  to  $\tilde{Q}'$ . Two of  $q$ 's vertices are assigned different colors at the reattachment points, the endpoints of  $d$ , and the other two vertices of  $q$  can be assigned the remaining two colors.



**Fig. 2.18.** Orthogonal version of Fig. 1.2 establishes  $\lfloor n/4 \rfloor$  necessity.



**Fig. 2.19.** A 4-colorable graph derived from a quadrilateralization by adding all quadrilateral diagonals.

Since the quadrilaterals cover  $P$  and are convex, placing guards at the vertices assigned the least frequently used color will cover the interior of  $P$ . As this color must be used no more than  $\lfloor n/4 \rfloor$  times, the theorem is established.  $\square$

Note that the quadrilaterals clipped in this proof are “orthogonal ears”; thus every orthogonal polygon has at least two such ears, providing an orthogonal counterpart to Theorem 1.3.

We will see in Section 2.5 below that the powerful quadrilateralization theorem is not necessary to prove  $\lfloor n/4 \rfloor$  sufficiency, but it does seem to be an essential tool in many other proofs. We now turn to an algorithm for constructing a convex quadrilateralization.

## 2.3. SACK'S QUADRILATERALIZATION ALGORITHM

### 2.3.1. Introduction

The proof presented in the preceding section does not immediately lead to an efficient algorithm. The first such algorithm is due to Sack (1984), and although it has been superseded to a certain extent by Lubiw's algorithm (Section 2.4), it remains interesting because it is an exact parallel of Lee and Preparata's monotone partitioning algorithm (Section 1.3.2). In addition, when supplemented by proofs of correctness, it can be seen as an alternative proof of the quadrilateralization theorem. Most of the proofs will only be sketched in this section; the reader is referred to Sack's thesis (Sack 1984) for more thorough proofs.

His algorithm factors the problem into three subproblems: partitioning into monotone polygons, quadrilateralization of monotone polygons, and quadrilateralization of “pyramids.” This latter problem (first analyzed in Sack and Toussaint (1981)) is solved repeatedly during the quadrilateralization of a monotone polygon. We will describe the algorithm “bottom up,” starting with pyramids.

### 2.3.2. Pyramid Quadrilateralization

Define a (vertical) *histogram* as an orthogonal polygon with one horizontal edge (the base) equal in length to the sum of the lengths of all the other horizontal edges (Edelsbrunner *et al.*, 1984). A (vertical) *pyramid* is then defined as a vertical histogram that is monotone with respect to the vertical direction.<sup>4</sup> It is easily seen that a pyramid must consist of two (perhaps empty) “staircases” as illustrated in Fig. 2.20. It is not surprising that these highly specialized polygons are easy to quadrilateralize.

Label the reflex vertices on the left staircase  $l_1, \dots, l_{a-1}$  from top to

4. This is equivalent to being horizontally *convex*, which requires that the polygon meet every horizontal line in a single segment.

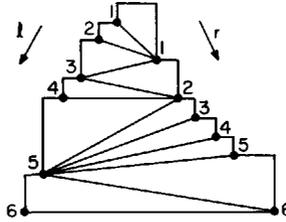


Fig. 2.20. Quadrilateralization of a pyramid by list merging.

bottom, and label the left endpoint of the pyramid base  $l_a$ . Similarly let  $r_1, \dots, r_{b-1}$  be the reflex vertices in the right staircase sorted by height, and let  $r_b$  be the right endpoint of the base.

Now merge the two sorted lists of vertices; for the example in Fig. 2.20 the result is

$$l_1 l_2 r_1 l_3 l_4 r_2 r_3 r_4 r_5 l_5 r_6 l_6$$

Finally, for each vertex  $l_i$  in the list, draw a diagonal to the next  $r_j$  in the list, and similarly for each  $r_j$ , draw a diagonal to the next  $l_i$ . We will show that this procedure quadrilateralizes the pyramid in linear time. First we present a more formal description of the algorithm.

*Algorithm 2.1* (Pyramid Quadrilateralization)

- (1) Form two lists  $l_a, \dots, l_2, l_1$  and  $r_b, \dots, r_2, r_1$ , sorted bottom to top, of the reflex vertices in the left and right staircases, with  $l_a$  and  $r_b$  the endpoints of the base.
- (2) Merge these two sorted lists; call the result  $L$ .
- (3)  $target \leftarrow \text{head}(L)$ .  
 $L \leftarrow \text{tail}(L)$ .  
**while**  $L$  not empty **do**  
**begin**  
**if**  $\text{head}(L)$  on same staircase as  $target$   
**then**  $target \leftarrow \text{head}(L)$   
**else** output diagonal ( $\text{head}(L)$ ,  $target$ )  
 $L \leftarrow \text{tail}(L)$ .  
**end**

It is clear that this algorithm only requires linear time: the stairway lists can be constructed (1) in linear time, merging (2) takes only linear time, and the **while** loop (3) removes an element with each pass, and so also consumes just linear time.

We now turn to correctness. Each pair of adjacent reflex vertices on one staircase is connected to a common vertex on the other staircase. Thus the pieces of the induced partition are quadrilaterals. All the convex vertices are included in these quadrilaterals, and every reflex vertex is the source for a diagonal. Thus the quadrilaterals cover the polygon. It only remains to show that the quadrilaterals are convex.

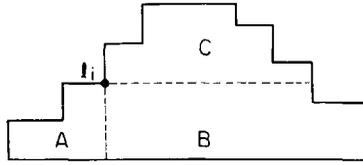


Fig. 2.21.  $l_i$  is connected to a vertex in  $B$ .

Let  $l_i$  be a reflex vertex on the left stair, as illustrated in Fig. 2.21, and let  $r_j$  be one of the target vertices on the right stair to which it is connected by the algorithm. Since  $r_j$  is on the right stair, it cannot be in region  $A$  of the figure. Since  $r_j$  is later in the list  $L$  than  $l_i$ , it is lower and cannot be in region  $C$ . Thus  $r_j$  is in region  $B$ , and so the concavity at  $l_i$  is “broken.” Since every reflex concavity is broken, the resulting quadrilaterals are all convex. This completes the proof of correctness.

In the next section we will need a slightly stronger result. Define a *pseudo-pyramid*<sup>5</sup> as a monotone pyramid whose horizontal step edges may be slanted—that is, non-horizontal but sloping upwards maintaining monotonicity—and whose base edge may be sloped in either direction as long as both endpoints are below the lowest reflex vertex; see Fig. 2.22. It is easy to establish that Algorithm 2.1 also works for pseudo-pyramids, and we will use this result in the next section.

### 2.3.3. Orthogonal Monotone Quadrilateralization

Sack’s monotone quadrilateralization algorithm makes a single pass over the polygon from top to bottom (the polygon is assumed to be monotone with respect to the vertical). Let the horizontal edges encountered in such a pass be  $e_1, \dots, e_n$ , with  $e_1$  highest. The algorithm will sometimes treat one of the diagonals it outputs as one of the  $e_i$ , a slanted horizontal edge. In either case, the action taken depends on whether  $e_i$  is a top edge or a bottom edge. Top edges are pushed onto a stack, forming the non-vertical edges of a pseudo-pyramid. Bottom edges cause one or two diagonals to be output, and perhaps the pyramid algorithm to be called to quadrilateralize the pyramid contained in the stack.

Edges will be identified as touching the left, the right, or both chains. Of

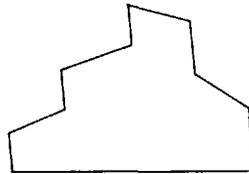


Fig. 2.22. A pseudo-pyramid.

5. Sack’s nomenclature is “worn-pyramid.”

two edges  $a$  and  $b$  touching the left chain only,  $a$  is said to extend *further inside* than  $b$  if  $a$ 's right endpoint is right of  $b$ 's right endpoint; the term is similarly defined for edges touching the right chain only.

The algorithm is presented next, followed by an example and analysis.

*Algorithm 2.2* (Monotone Quadrilateralization).

Sort horizontal edges from top to bottom; let  $e_1, \dots, e_n$  be the result.

Push  $e_1$  onto stack  $S$ .

$i \leftarrow 2$

**while**  $S$  not empty **or**  $i \neq n$  **do**

**begin**

**if**  $i = n$  **then**

**begin**

Push  $e_i$  on  $S$ .

Call pyramid algorithm for  $S$ .

exit.

**end.**

**case**  $e_i$

(1) top segment: Push  $e_i$  on  $S$ .

$i \leftarrow i + 1$

(2) bottom segment:  $e \leftarrow \text{Pop } S$ .

**case**  $e$

(A)  $e$  and  $e_i$  touch same chain:

Join other ends of  $e$  and  $e_i$  by diagonal  $d$ .

**if**  $e$  touches both chains **or**  $e$  extends further inside than  $e_i$

**then**

**begin**

Push  $d$  on  $S$ .

$i \leftarrow i + 1$

**end**

**else**  $e_i \leftarrow d$  {NB:  $i$  is not incremented}

(B)  $e$  and  $e_i$  are on opposite chains:

Join  $e$  and  $e_i$  with diagonals  $d_1$  (higher) and  $d_2$ .

Push  $d_1$  on  $S$ .

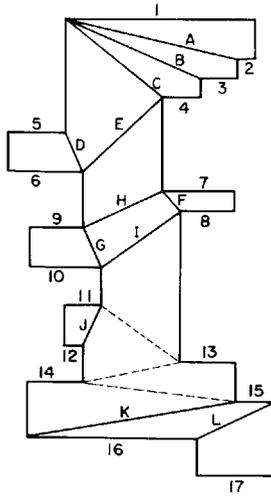
Call pyramid algorithm for  $S$ ;

Push  $d_2$  on  $S$ ;

$i \leftarrow i + 1$

**end**

An example illustrating the execution of the algorithm is shown in Fig. 2.23; Table 2.1 shows the stack after processing  $e_i$ . Note that an upside-down pyramid is quadrilateralized directly, without calling the pyramid algorithm (e.g.,  $e_1$  through  $e_4$ ); only pyramids with bases at the bottom are pushed onto the stack (e.g., diagonals I through K). This asymmetry makes it clear that the upright pyramids do not have to be handled with a special algorithm, but factoring the problem this way does make it easier to understand (and prove correct).



**Fig. 2.23.** A partition produced by the monotone quadrilateralization algorithm.

**Table 2.1**

$e_i$	$S$
1	1
2	A
3	B
4	C
5	5C
D	C
6	E
7	7E
8	FE
9	9FE
10	9FE
G	FE
	HE (pyr)
11	I
12	11I
13	JI
14	13JI
15	1413JI
16	151413JI
	K 151413JI (pyr)
17	L
	17L (pyr)

We now sketch an inductive proof of correctness; a more detailed proof may be found in Sack (1984). The induction hypothesis is the conjunction of these three statements:

- (1) The stack  $S$  contains only the non-vertical edges of an upright pseudo-pyramid.
- (2) No two edges in  $S$  contain points with the same  $y$ -coordinate.
- (3) The next edge examined by the **while** loop,  $e_i$ , is connected by a vertical edge to the lowest edge of  $S$  on the same chain.

These statements are trivially true initially. If  $e_i$  is a top edge (Case 1), the stack is augmented, and by (3), this is a proper augmentation of a pyramid. Clearly statement (3) remains true, either because  $e_{i+1}$  is on the chain opposite that of  $e_i$ , or because  $e_{i+1}$  is directly connected to  $e_i$  by a vertical edge.

If  $e_i$  is a bottom edge on the same chain as the stack top  $e$  (Case 2A), then regardless of whether  $e$  touches both chains, or extends further inside than  $e_i$ , (3) guarantees that adding a single diagonal  $d$  will cut off a convex quadrilateral. If  $e$  extends further inside, then  $d$  represents a “wearing down” of the pyramid, and is properly pushed. Otherwise  $d$  acts just like a bottom edge. It is easily verified that in either case the truth of statements (2) and (3) is maintained.

If  $e_i$  is a bottom edge on the chain opposite that touched by the stack top  $e$  (Case 2B), then by properties (2) and (3), the region from the top of  $e$  to  $e_i$  is empty, and  $e_i$  and  $e$  can be connected by a pair of diagonals. The upper diagonal connects the two chains of the pyramid in  $S$ , and the stack is emptied by a call to Algorithm 2.1. The lower diagonal forms a new pyramid top.

This completes the proof of correctness. The algorithm clearly requires just linear time: sorting the horizontal edges can be accomplished by a linear merge of the edges in the two monotone chains, and the **while** loop makes at most as many passes as there are edges and diagonals, which is also linear. Finally, the pyramid algorithm is itself linear.

As with the generalization of pyramid to pseudo-pyramid, it will be important in the next section to generalize the class of polygons for which Algorithm 2.2 applies from orthogonal monotone to *pseudo-monotone* polygons. These are polygons that are:

- (1) monotone with respect to the vertical,
- (2) composed of vertical edges alternating with non-vertical edges, which must be either horizontal or upwardly slanting to satisfy (1),
- (3) and such that the *shadow* of a slanted edge contains no vertices, where the shadow of a slanted edge is defined to be the set of points of the polygon visible to  $e$  by a horizontal line segment that is nowhere exterior to the polygon, but not including the endpoints of  $e$ .

This third requirement (similar to clause (2) of the induction hypothesis) is

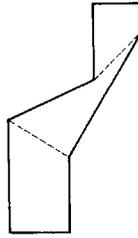


Fig. 2.24. A non-quadrilateralizable monotone polygon.

needed to avoid polygons such as the one shown in Fig. 2.24 that do not admit a convex quadrilateralization.

It is easily established that Algorithm 2.2 will work on pseudo-monotone polygons. The next section presents an algorithm for partitioning an orthogonal polygon into pseudo-monotone pieces.

#### 2.3.4. Partitioning into Monotone Polygons

Define a *bottom peak* to be a bottom horizontal edge whose endpoints are both reflex vertices; define a *top peak* similarly for top edges. Then it is clear that an orthogonal polygon fails to be monotone with respect to the vertical direction precisely when it contains top or bottom peaks: these peaks play the role of up and down cusps in a general polygon (Section 1.3.2). Sack's algorithm is exactly analogous to Lee and Preparata's: the polygon is cut at each bottom peak by adding diagonals to the closest horizontal edge above it, and similarly for top peaks. The geometric lemma that permits the diagonals to be added is as follows.

**LEMMA 2.10.** Let  $ab$  be a bottom peak, and let  $cd$  be the lowest horizontal edge above  $ab$  that is partially visible to either  $a$  or  $b$ . Then:

- If  $cd$  is a top edge, then the rectangle bounded by  $ab$ ,  $cd$ , and the facing edges of this neighboring pair (see Lemma 2.2), is empty. See Fig. 2.25a.
- If  $cd$  is a bottom edge, then  $ab$  and  $cd$  do not overlap horizontally, and the rectangle bounded by  $ab$ , the visible endpoint of  $cd$ , and the "facing" vertical edges, is empty. See Fig. 2.25b.

*Proof.* (Sketch). Part (a) is equivalent to Lemma 2.2. For part (b), if  $cd$

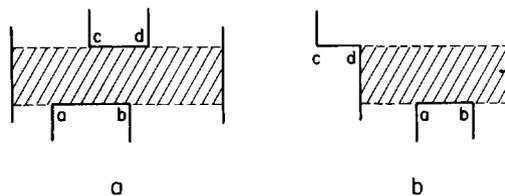


Fig. 2.25. Empty rectangles above bottom peaks.

and  $ab$  overlapped horizontally, then there would have to be an edge lower than  $cd$  and visible to  $a$  or  $b$ , contradicting the assumption that  $cd$  is the lowest. An argument similar to that used in Lemma 2.2 can be used to establish the emptiness of the rectangle illustrated in Fig. 2.25b.  $\square$

Because the algorithm is so similar to Lee and Preparata's Algorithm 1.2, we will only present a high-level version here; a detailed version may be found in Sack (1984). The reason that Lee and Preparata's algorithm cannot be used without modification is that the monotone pieces resulting from their algorithm do not necessarily satisfy condition (3) in the definition of "pseudo-monotone."

*Algorithm 2.3* (Pseudo-Monotone Partitioning of an Orthogonal Polygon). Sort the horizontal edges, top to bottom; let  $e_1, \dots, e_n$  be the result.

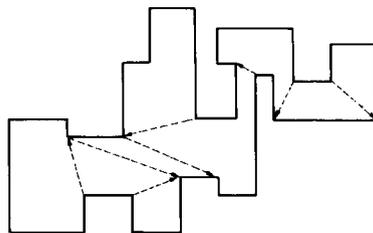
```

for  $i = 2$  to  $n$  do
  if  $e_i$  is a bottom peak
    then  $e \leftarrow$  lowest horizontal or slanted edge above and visible to  $e_i$ .
    if  $e$  is a top edge
      then Join  $e$  and  $e_i$  into a convex quadrilateral
      else Join the visible endpoint of  $e$  to the closest endpoint of  $e_i$ .
      {Data structure manipulation here.}
    else if  $e_i$  is a top peak then {Similar to above.}
  
```

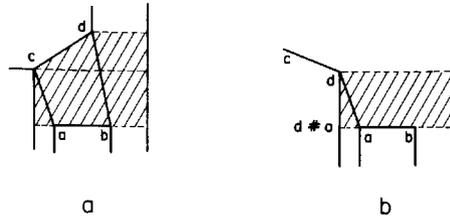
This algorithm makes a single pass over the polygon. The algorithm could be simplified by handling the bottom peaks on a top-to-bottom pass and the top peaks on a bottom-to-top pass (as is done in Algorithm 1.2), but we will maintain this more complicated form to simplify the correctness discussion. Note that the step that joins  $e$  and  $e_i$  into a convex quadrilateral is a special case of the first reduction (of neighboring edges that do not form a tab) in the Kahn, Klawe, Kleitman proof presented in the previous section.

An example of the partition created by the algorithm is shown in Fig. 2.26, where upward pointing arrows resolve bottom peaks, and downward arrows resolve top peaks.

We now argue that this algorithm partitions an orthogonal polygon into pseudo-monotone pieces in  $O(n \log n)$  time. Assume as an induction hypothesis that all bottom and top horizontal peaks above a certain height



**Fig. 2.26.** A partition produced by the monotone partition algorithm.



**Fig. 2.27.** The diagonals chosen by the algorithm pass through empty (shaded) regions.

have been resolved by the algorithm, and that the resulting monotone chains above this height satisfy the pseudo-orthogonal criteria:

- (1) Vertical and slanted edges alternate.
- (2) The shadow of each slanted edge contains no vertices.

Consider now the next peak encountered, and assume it is a bottom peak  $e_i = ab$  (the argument for a top peak is slightly different and will not be presented).

Let  $cd$  be edge  $e$  selected by the algorithm. There are two cases, illustrated in Fig. 2.27.

*Case A* ( $cd$  is a top segment). Since  $cd$  satisfies condition (2) above, both  $c$  and  $d$  project horizontally to the same vertical edge. By a slight modification of Lemma 2.10, the entire shaded region of Fig. 2.27a is empty, and  $abcd$  forms an internal convex quadrilateral. Diagonals  $ac$  and  $bd$  satisfy both properties (1) and (2) of the induction hypothesis. Compare Fig. 2.8: in Fig. 2.27a,  $ab$  and  $cd$  also are non-tab neighboring edges.

*Case B* ( $cd$  is a bottom segment). The situation is illustrated in Fig. 2.27b. Assume that the diagonal added by the algorithm is  $ad$ . This diagonal is easily seen to satisfy condition (2) of the induction hypothesis, but it does not satisfy condition (1), as now  $ad$  and  $dc$  are consecutive slanted edges. The solution is to replace  $a$  by  $d\#a$  in the upper polygon. Then the upper polygon satisfies conditions for a pseudo-orthogonal monotone polygon, and can therefore be quadrilateralized by Algorithm 2.2. It is possible to show (see Sack (1984)) that this algorithm will not use  $d\#a$  as the endpoint of a diagonal, and that the unique quadrilateral that includes this vertex remains convex when  $d\#a$  is replaced by  $a$ .

This completes our sketch of the correctness of Algorithm 2.3. Its time complexity is  $O(n \log n)$  for the same reasons Lee and Preparata's has this bound: the initial sorting of the horizontal edges requires  $O(n \log n)$ , and there are  $O(n)$  insertions and deletions into a dictionary data structure. I believe that Tarjan and Van Wyk's trapezoidalization algorithm mentioned in Section 1.3.2 can be used to improve the speed of Sack's algorithm to  $O(n \log \log n)$ .

## 2.4. LUBIW'S PROOF AND ALGORITHM

### 2.4.1. Introduction

In this section we present a clever and succinct proof of the convex quadrilateralization theorem due to Lubiw (1985). Her proof leads rather directly to another  $O(n \log n)$  algorithm. In fact, we present two proofs: one for orthogonal polygons without holes, and one for those with holes. The latter obviously encompasses the former, but the proof for polygons without holes is so elegant that it deserves separate consideration.

Both proofs have the same structure, and depend on the following observation. From a quadrilateralization of an orthogonal polygon (with or without holes), remove one quadrilateral. What remains are perhaps several polygons each of which is quadrilateralizable. But these polygons are not (in general) themselves orthogonal. Thus there is a broader class of polygons beyond orthogonal to which the theorem applies; no one has characterized this class to date. Lubiw identified such classes for orthogonal polygons both with and without holes. For each member of the class, there is a quadrilateral whose removal results in smaller polygons within the same class. The quadrilateralization theorem follows immediately by induction.

Kahn, Klawe, and Kleitman's proof (Section 2.3) stays within the class of orthogonal polygons at all times; Lubiw's proof starts with a wider class more suited to the removal of quadrilaterals. The result is a simpler proof.

### 2.4.2. Orthogonal Polygons without Holes

Define the dual of a quadrilateralization of a polygon as in Section 2.2.2: a graph with a node associated with each quadrilateral, and an arc between two nodes if the corresponding quadrilaterals share a diagonal.<sup>6</sup> As previously mentioned, the dual graph of a quadrilateralization of a orthogonal polygon without holes is a tree, for the same reason that the dual of a triangulation of a polygon without holes is a tree (Lemma 1.3). Removal of one quadrilateral from a quadrilateralization of a polygon therefore disconnects the polygon into quadrilateralizable pieces, each of which is orthogonal except for one slanted edge. This observation motivates the following definition.

A *1-orthogonal polygon* is a polygon of no holes with a distinguished edge  $e$  called the *slanted edge*, such that the polygon satisfies four conditions:

- (1) There are an even number of edges.
- (2) Except for possibly  $e$ , the edges are alternately horizontal and vertical in a traversal of the boundary.
- (3) All interior angles are less than or equal to  $270^\circ$ .
- (4) The nose of the slanted edge contains no vertices.

6. These graphs will be studied further in the next chapter.

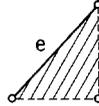


Fig. 2.28. The nose of a slanted edge.

The *nose* of a slanted edge  $e$  is the right triangle toward the inside of the polygon whose hypotenuse is  $e$ ; the nose includes the interior of  $e$  but excludes the remainder of the boundary. See Fig. 2.28. Clearly the requirement that the nose not contain vertices implies that it is completely empty, since all other edges must be vertical or horizontal and could not intersect the nose without including their endpoint.

An orthogonal polygon is 1-orthogonal, where  $e$  may be any edge. Violation of any of the four conditions can lead to non-quadrilateralizable polygons, as is illustrated in Fig. 2.29.

**THEOREM 2.3** [Lubiw 1985]. Any 1-orthogonal polygon  $P$  is convexly quadrilateralizable.

*Proof.* The proof is by induction. If  $P$  has just four edges, then it has (perhaps after rotation by  $90^\circ$ ) two horizontal edges and one vertical edge by (2). Regardless of the orientation of the slanted edge,  $P$  must be convex. This establishes the basis of the induction.

Assume now that  $P$  has more than four edges. We will show that there exists a *removable quadrilateral*, a convex quadrilateral whose removal disconnects  $P$  into smaller 1-orthogonal polygons.

Properties (1) and (2) jointly imply that the two edges adjacent to the slanted edge  $e = ab$  are either both horizontal or both vertical. The interior angle requirement (3) then implies that one of the two situations illustrated in Fig. 2.30 holds (perhaps after rotation and/or reflection). In both cases, a

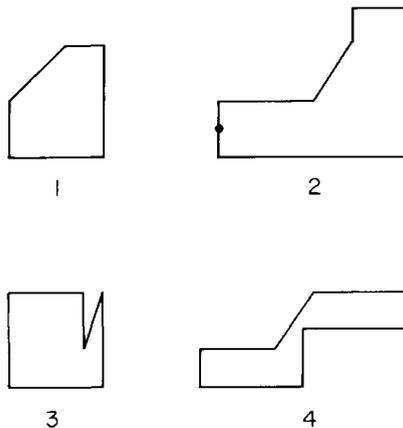
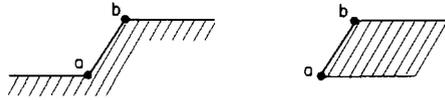
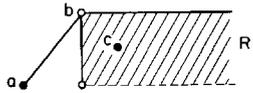


Fig. 2.29. Four unquadrilateralizable polygons, each violating one of the 1-orthogonal conditions.



**Fig. 2.30.** The slanted edge  $ab$  in standard orientation; the shading indicates the interior of the polygon.

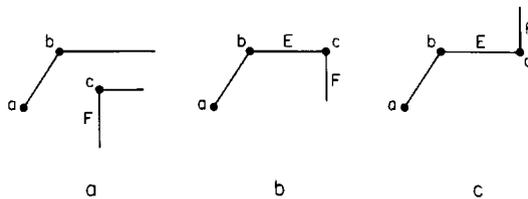
top edge has  $b$  as its left endpoint. The removable quadrilateral has  $a$  and  $b$  as two of its vertices. We now specify the other two vertices  $c$  and  $d$ . Vertex  $c$  is the leftmost then highest vertex within the region  $R_1$  shown in Fig. 2.31.  $R_1$  is closed along the left and top, does not include the two left corners, and is open elsewhere. Since  $R_1$  contains at least the vertex at the other end of the top edge  $E$  incident to  $b$ ,  $c$  must exist. If  $c$  is not this other endpoint, then because it is leftmost, it is at the top of a vertical edge. Thus it is either at a corner as in Fig. 2.32a, or at the other end of  $E$  as in Figs. 2.32b and 2.32c.



**Fig. 2.31.** Definition of vertex  $c$  and region  $R_1$ .

Corner  $d$  of the removable quadrilateral is specified according to two cases. Define region  $R_2$  as shown in Fig. 2.33. It is closed on the top and right, excludes  $a$  and  $c$ , and is otherwise open. Define  $d$  to be, if it exists, the highest then rightmost vertex in  $R_2$ . If either of the situations illustrated in Fig. 2.32a or 2.32b obtains, then  $R_2$  contains at least the vertex at the lower end of the vertical edge  $F$  incident to  $c$ , and so  $d$  is defined. In these cases, if  $F \neq cd$ , then because  $d$  is highest, it is at the top of a vertical edge, and because rightmost, it is at the right of a horizontal edge, and thus at a corner.

If the situation shown in Fig. 2.32c holds, however,  $d$  may be the left endpoint of a bottom edge  $G$  in  $R_2$  (marked as  $d'$  in Fig. 2.34a) or  $R_2$  may contain no vertex at all (Fig. 2.34b). In either of these situations, define the region  $R_3$  as illustrated; it is closed on the left and bottom, excludes the two left corners, and is otherwise open. Now define  $d$  to be the leftmost then lowest vertex in  $R_3$ . Since  $R_3$  includes at least the right endpoint of  $G$ ,  $d$  must exist.



**Fig. 2.32.** Either  $c$  is at a corner (a) or at the other end of  $E$  (b and c).

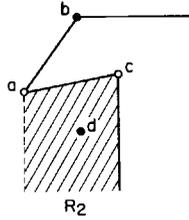


Fig. 2.33. Definition of vertex  $d$  and region  $R_2$ .

Having now defined vertices  $c$  and  $d$  under all circumstances, remove the quadrilateral  $abcd$ . First note that it is convex: the angle at  $b$  is convex because  $c$  is in  $R_1$ ; the angle at  $a$  is convex because  $d$  is in  $R_2$  or  $R_3$ ; the angles at  $c$  and  $d$  are convex both when  $R_2$  and  $R_3$  are applicable.

Removal of  $abcd$  potentially leaves three polygonal regions, bounded by  $bc$ ,  $cd$ , and  $ad$ . Each of these diagonals is clearly the only slanted edge in its respective polygon; thus condition (2) in the definition of 1-orthogonal is satisfied for each. Condition (3), that all angles are less than or equal to  $270^\circ$ , is easily seen to hold by examining the cases considered above, which are gathered in Fig. 2.35. That the nose of each slanted edge is empty (condition (4)) is guaranteed by choice of  $c$  as leftmost then highest, and the similar constraints on  $d$ . These constraints force the shaded regions in Fig. 2.35 to be empty. In all cases, the nose of each slanted edge is empty.

Finally, to prove that each remaining polygon has an even number of edges (condition (1)), we use the following clever counting argument of Lubiw. Define the endpoints of top, bottom, left, and right edges as partitioned into two types as specified in Fig. 2.36a. Endpoints of a slanted edge are given a type by the adjacent horizontal or vertical edges. It should be clear that each vertex of a 1-orthogonal polygon is assigned an unambiguous type by this scheme since, for example, a convex corner formed by a bottom and a left edge can only be a type 1 vertex. In any traversal of a 1-orthogonal polygon, the vertices alternate in type. Now note that  $a$  is type 1 and  $b$  is type 2 (Fig. 2.30); the cases shown in Fig. 2.35 show that  $c$  is always type 1 and  $d$  always type 2. Thus each of the diagonals  $bc$ ,  $cd$ , and  $ad$  has both type 1 and type 2 endpoints (see Fig. 2.36b), and

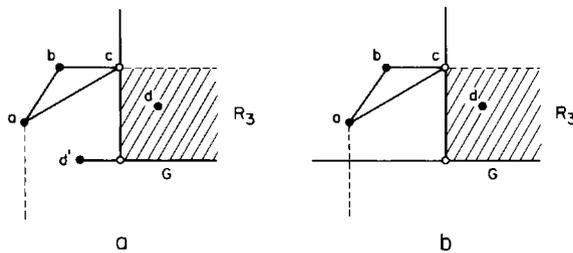
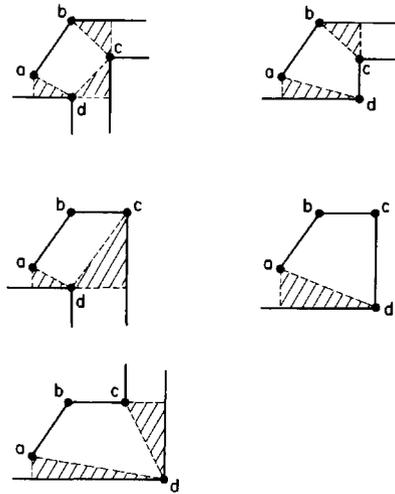


Fig. 2.34. Definition of vertex  $d$  and region  $R_3$ .



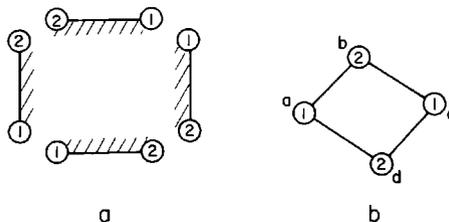
**Fig. 2.35.** The noses of the diagonals of the removable quadrilateral are all empty (shaded regions).

therefore the polygon pieces bounded by these diagonals maintain the type  $1/2$  alternation. Thus they must have an even number of vertices.

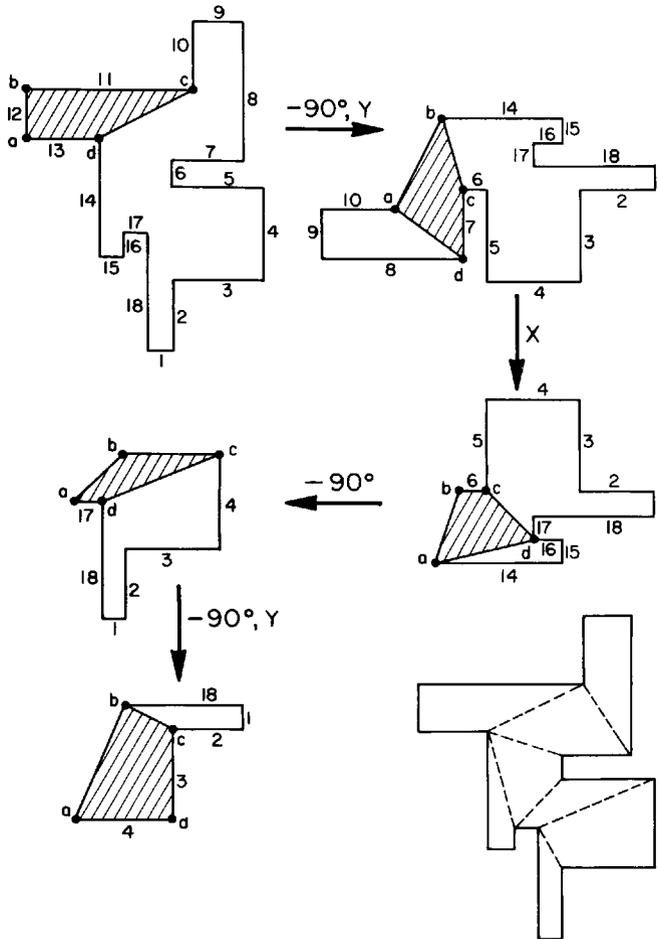
We have therefore shown that each of the pieces remaining is a 1-orthogonal polygon. Clearly they have fewer vertices than  $P$ , so the induction hypothesis guarantees they are quadrilateralizable. Joining their quadrilateralizations to  $abcd$  yields a quadrilateralization of  $P$ , establishing the theorem.  $\square$

An example of a quadrilateralization obtained by repeatedly removing the removable quadrilateral defined in this theorem is shown in Fig. 2.37. In the original polygon, edge 12 is chosen as the (degenerate) slanted edge. Rotations and reflections are often necessary to orient the slanted edge to match Fig. 2.30, but the procedure is entirely deterministic.

We now turn now to the more difficult case of orthogonal polygons with holes.



**Fig. 2.36.** Vertex types; the shading in (a) represents the polygon interior.



**Fig. 2.37.** Quadrilateralization by repeated deletion of a removable quadrilateral. The notation “ $-90^\circ, Y$ ” means that the figure is rotated  $90^\circ$  clockwise and then reflected in the  $Y$ -axis; the remaining arrow labels may be interpreted similarly.

### 2.4.3. Orthogonal Polygons with Holes

Because the dual of a quadrilateralization of a polygon with holes is not a tree, the removal of one quadrilateral may introduce more than one slanted edge into the remainder. For example, removal of any quadrilateral from the polygon shown in Fig. 2.38 introduces two slanted edges. Thus the class of 1-orthogonal polygons will not be sufficient to prove the more general theorem by the same technique.

Here Lubiw uses the same basic insight (independently) employed by Sack with his pseudo-pyramids and pseudo-monotone orthogonal polygons: if the horizontal edges of an orthogonal polygon are slanted, and some

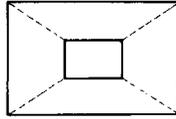


Fig. 2.38. Removal of any quadrilateral produces two slanted edges.

further conditions are satisfied, then the resulting polygon is still quadrilateralizable. This leads to the following definition:

A *pseudo-orthogonal* polygon is one that satisfies three conditions:

- (1) Every other edge in a traversal is vertical. The other edges are called *slanted*; note a slanted edge may be horizontal.
- (2) All interior angles are less than or equal to  $270^\circ$ .
- (3) The shadow of any slanted edge contains no vertices.

The *shadow* of an edge  $e$  is, as defined in Section 2.2, the horizontal projection of  $e$ . For the precise purposes of her proof, Lubiw defines the shadow to be open on the bottom if  $e$  is a left edge, and open on the top if a right edge, and in either case not including  $e$ 's endpoints; see Fig. 2.39. Closing the open boundaries does not lead to unquadrilateralizable polygons, but the definition as stated leads to precise meshing of regions in the proof. A horizontal edge is defined to have no shadow; thus all orthogonal polygons (with or without holes) are pseudo-orthogonal.



Fig. 2.39. The shadow of an edge; the shading indicates the interior of the polygon

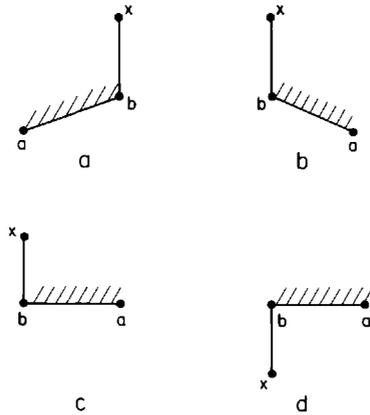
**THEOREM 2.4** [Lubiw 1985]. A pseudo-orthogonal polygon  $P$  is convexly quadrilateralizable.

*Proof.* The proof is by induction. If  $P$  has just four edges, then two must be horizontal by (1), and regardless of the slant of the slanted edges,  $P$  is convex. This establishes the basis step.

If  $P$  has more than four edges, then we show that  $P$  has a removable quadrilateral: one whose removal leaves smaller polygons which are themselves pseudo-orthogonal. Toward this end, define the *upper neighbor*  $u$  of a vertex  $v$  as the lowest then rightmost vertex above  $v$  that is visible to  $v$ , but not connected to  $v$  by a slanted edge, and strictly above  $v$  if  $v$  is on a top edge.<sup>7</sup> Not every vertex has an upper neighbor: for example, a convex corner at the junction of a top edge and a right edge has none.

We can now specify a removable quadrilateral. Throughout the remainder of the proof we will use  $x < y$  to mean that  $x$  is lower than  $y$ , or they

<sup>7</sup> These latter qualifications will not be needed in the proof, but only in the algorithm correctness discussion to follow.



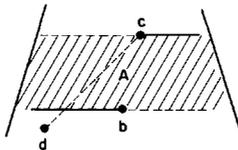
**Fig. 2.40.**  $b$  always has an upper neighbor; the shading indicates the interior of the polygon.

have the same height and  $x$  is right of  $y$ . Let  $ab$  with  $a < b$  be a bottom edge, and let  $b$ 's upper neighbor be  $c$ . If  $c$  is on a top edge  $cd$ , then  $ab$  and  $cd$  define a removable quadrilateral.

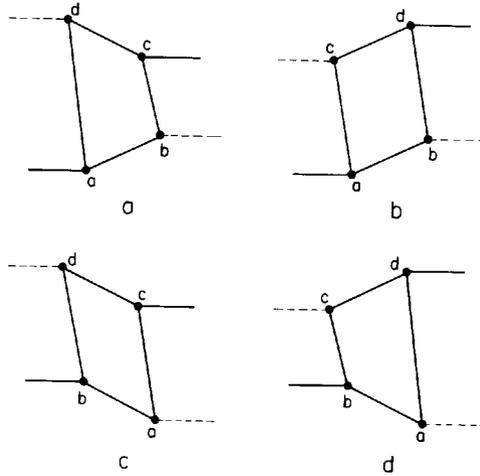
We must establish that this quadrilateral always exists. First we show that  $b$  always has an upper neighbor. If  $ab$  is not horizontal, then the vertical edge incident to  $b$  must be above  $b$  to satisfy the  $270^\circ$  requirement (2) (see Figs. 2.40a and 2.40b). The other endpoint of this edge guarantees that  $b$  has an upper neighbor. If  $ab$  is horizontal, then  $b$  is at the left. If the angle at  $b$  is convex (Fig. 2.40c) then the situation is as above. If the angle is reflex (Fig. 2.40d), then there clearly must be at least two vertices above and visible to  $b$ , so again  $b$  has an upper neighbor.

Second, we need to show that there is an  $ab$  as specified with  $b$ 's upper neighbor on a top edge. Let  $ab$  with  $a < b$  be the bottom edge with  $b$  maximum among all bottom edges with respect to " $<$ ". Thus  $b$  is the uppermost upper endpoint of all the bottom edges. Then  $b$ 's upper neighbor  $c$  cannot be on a bottom edge. Since every vertex has a bottom or top (i.e., a slanted) edge incident to it by (1),  $c$  must be on a top edge.

Having shown that the quadrilateral always exists, we now show that it is removable. Let  $A$  be the set of points vertically between  $b$  and  $c$ , excluding those right of  $b$  and left of  $c$ , as illustrated in Fig. 2.41. Since  $c$  is the upper neighbor of  $b$ ,  $A$  contains no vertices. Let  $d$  be the other end of the top edge containing  $c$ . If  $d < c$ , then it must be that  $d < b$  to avoid  $A$ . But this would put  $b$  in the shadow of  $dc$ , violating condition (3). Thus we must have



**Fig. 2.41.** The region  $A$  between  $b$  and its upper neighbor  $c$ .

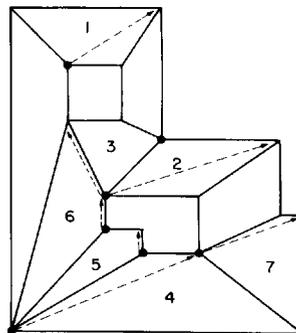


**Fig. 2.42.** The removable quadrilateral in all its orientations. The horizontal solid and dashed edges represent the closed and open boundaries of  $B$ .

$c < d$ . This leads to four cases, as illustrated in Fig. 2.42. Let  $B$  be the union of  $A$  and the shadows of  $ab$  and  $cd$ . Then  $B$  is empty and so the quadrilaterals illustrated in the figure are each internal to  $P$ .

Next we show that the removable quadrilateral is convex. If  $ab$  is not horizontal, then the angle at  $a$  is clearly less than  $180^\circ$ , and the angle at  $b$  is less than  $180^\circ$  because the vertical edge incident to  $b$  must extend upwards to satisfy (2). If  $ab$  is horizontal, then both angles are less than  $180^\circ$  since  $c$  and  $d$  are both above  $ab$ . Exactly analogous arguments show that the angles at  $c$  and  $d$  are less than  $180^\circ$ , thereby establishing that the quadrilateral is convex.

Removal of the quadrilateral introduces two new slanted edges in each case of Fig. 2.42. Notice that in each case, the shadow of these slanted



**Fig. 2.43.** Quadrilateralization by repeated deletion of removable quadrilaterals. The quadrilaterals are numbered in the order in which they are removed. The dashed arrows represent the upper neighbor relation.

edges is in  $B$  and so is empty, thus satisfying condition (3). Since vertical edges are incident to  $a$ ,  $b$ ,  $c$ , and  $d$ , the alternation of vertical and slanted is maintained (1). Finally, since angles less than or equal to  $270^\circ$  are being subdivided by the removal, condition (3) holds for the remainder. Thus the removal leaves pseudo-orthogonal pieces. Applying the induction hypothesis to these pieces, and merging the resulting quadrilateralizations with the removed quadrilateral, provides a quadrilateralization for  $P$  and establishes the theorem.  $\square$

An example of repeated application of the theorem to an orthogonal polygon with two holes is shown in Fig. 2.43. In the next section we show that this procedure can be implemented in  $O(n \log n)$  time.

#### 2.4.4. Lubiw's Algorithm

The proof in the preceding section leads to a surprisingly straightforward  $O(n \log n)$  algorithm for quadrilateralizing a pseudo-orthogonal polygon. First, the upper neighbors of all vertices are found in  $O(n \log n)$  time by a procedure nearly identical to Lee and Preparata's monotone partitioning algorithm (Section 1.3.2). Then quadrilaterals are removed as in Theorem 2.4 in linear time. The reason this simple approach works is the remarkable fact, proved by Lubiw, that the upper neighbors do not change when quadrilaterals are removed. If they did, recomputation would be necessary, and it would be very difficult to achieve  $O(n \log n)$  time.

Because the algorithm is so simple and similar to others discussed previously, it will only be sketched. However, its correctness will be proved in detail.

The first step of the algorithm is to find the upper neighbor of every vertex. Note the similarity between the connection of a vertex to its upper neighbor and the connection of a cusp to the closest visible vertex above it that is used in Lee and Preparata's monotone decomposition algorithm, or Sack's algorithm 2.3. Upper neighbors may be found in a single plane sweep in the same way that all upward cusps can be cut in a single sweep. The details differ in only minor ways; see Lubiw (1985) for a precise exposition.

The second step of the algorithm is to make a list  $L$  of all bottom edges, sorted bottom to top by their uppermost then leftmost endpoint. This can be accomplished during the same sweep that computes upper neighbors.

The final step of the algorithm is to remove the uppermost edge  $ab$  in  $L$ , identify the upper neighbor  $c$  of its upper endpoint  $b$ , and remove the removable quadrilateral specified in Theorem 2.4. This bottom edge  $ab$  satisfies the conditions of the theorem because it is the highest bottom edge. Moreover, the (at most) two new bottom edges introduced by the removal of the quadrilateral are above all other bottom edges in  $L$ , and so are placed the end of  $L$ .

What remains to be established is that the upper neighbors computed in

the first step remain valid throughout the execution of the algorithm. Lubiw's precise statement of this property is the following lemma.

**LEMMA 2.11.** Let  $P'$  result from  $P$  by repeated removal of quadrilaterals according to Theorem 2.4. Then if  $ab$  is a bottom edge of  $P'$  with  $a < b$ , the upper neighbor  $c$  of  $b$  in  $P'$  is the same as the upper neighbor of  $b$  in  $P$ .

*Proof.* Let  $bx$  be the vertical edge incident with  $b$  in  $P'$ . There are three possible relationships between  $a$ ,  $b$ , and  $x$ , as illustrated in Fig. 2.40:  $x$  may be below  $b$  only if  $ab$  is horizontal (Fig. 2.40d); otherwise it must be above (Figs. 2.40a, 2.40b, and 2.40c). The vertical edge incident to a vertex is only removed with a quadrilateral when the vertex itself is removed; thus  $bx$  is also the vertical edge incident to  $b$  in  $P$ .

Let  $c$  be the upper neighbor of  $b$  in  $P$ . Let  $A$  be the union of the two shadows of the chord  $bc$  towards the left and right. Because  $c$  is an upper neighbor of  $b$ ,  $A$  is devoid of vertices. The only way that  $c$  could *not* be the upper neighbor of  $b$  in  $P'$  is if the removal of some quadrilateral  $Q$  blocks the line of visibility  $bc$ . Since  $A$  is empty, such a  $Q$  must contain a vertex  $y \leq b$  and a vertex  $z \geq c$ . If  $b$  itself is not a vertex of  $Q$ , then it falls within a shadow of one of  $Q$ 's edges, inside the region  $B$  in the proof of Theorem 2.4, a contradiction. So  $b$  must be a corner of  $Q$ .

Consider now two cases. Suppose  $b$  is on the top edge forming  $Q$ . Then either  $b$  is at the right of a horizontal edge (Fig. 2.44a) or it is the highest endpoint of the top edge (Fig. 2.44b). Note that in the former case, the definition of upper neighbor implies that the left endpoint of the horizontal edge incident to  $b$  is *not* the upper neighbor of  $b$ , by requiring that the upper neighbor be strictly above  $b$  in this case. In either case,  $Q$  must lie below  $b$ , and there can be no vertex  $z$  of  $Q$ ,  $z \geq c$ .

Finally, suppose  $b$  is on a bottom edge forming  $Q$ . Then examination of Fig. 2.42 shows that  $b$  is either removed or converted to a lowest vertex of a bottom edge. And once it is this type of vertex, further quadrilateral removals can only reduce the internal angle, keeping it lowest on a bottom edge, contradicting our assumption that it is highest on a bottom edge in  $P'$ .

Thus no such obstructing  $Q$  will be removed by the algorithm, and therefore the upper neighbor relation remains fixed throughout.  $\square$

As with Sack's algorithm, I believe Tarjan and Van Wyk's trapezoidalization algorithm improves the speed of Lubiw's algorithm to  $O(n \log \log n)$ .

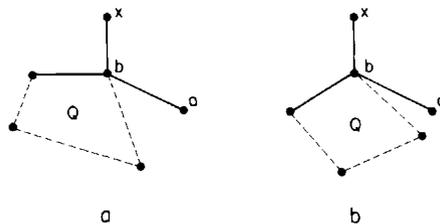


Fig. 2.44. Removal of  $Q$  cannot block  $b$  from its upper neighbor.

## 2.5. PARTITION INTO L-SHAPED PIECES<sup>8</sup>

Although the convex quadrilateralization theorem is deep and beautiful, it is not the only means of proving that  $\lfloor n/4 \rfloor$  guards are sufficient to cover an orthogonal polygon, nor the only avenue for placing the guards in  $O(n \log n)$  time. In this and the succeeding section, a different proof and algorithm are offered, based on a partition into L-shaped pieces rather than convex quadrilaterals. In this section we reprove the orthogonal art gallery theorem (Theorem 2.2), using the L-shaped partition introduced in O'Rourke (1983b).

### 2.5.1. Main Inductive Argument

The proof is phrased in terms of  $r$ , the number of reflex vertices of the orthogonal polygon, rather than  $n$ , the total number of vertices. This rephrasing is justified because there is a fixed relationship between  $r$  and  $n$ :

**LEMMA 2.12.** In an orthogonal polygon of  $n$  vertices,  $r$  of which are reflex,  $n = 2r + 4$ .

*Proof.* Let  $c$  be the number of vertices at which the internal angle is  $\pi/2$ ; clearly,  $n = c + r$ . Since the sum of the internal angles of a simple polygon is  $(n - 2)\pi$ , and since the angle at each reflex vertex is  $3\pi/2$ ,

$$(n - 2)\pi = c(\pi/2) + r(3\pi/2).$$

Solving for  $c$  and substituting into  $n = c + r$  yields  $n = 2r + 4$ .  $\square$

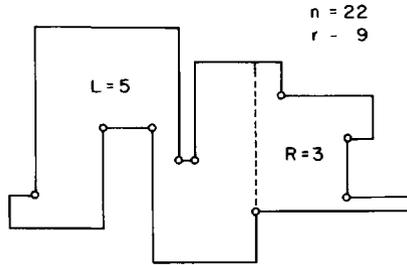
Since  $\lfloor n/4 \rfloor = \lfloor (2r + 4)/4 \rfloor = \lfloor r/2 \rfloor + 1$ , Theorem 2.2 can be stated as follows:

**THEOREM 2.5**  $\lfloor r/2 \rfloor + 1$  guards are necessary and sufficient to cover the interior of an orthogonal polygon of  $r$  reflex vertices.

The ‘‘comb’’ example (Fig. 2.18) establishes occasional necessity; an alternate sufficiency proof follows.

Define a *cut* of an orthogonal polygon as an extension of one of the two edges incident to a reflex vertex through the interior of the polygon until it first encounters the boundary of the polygon (see Fig. 2.45). A cut ‘‘resolves’’ its reflex vertex in the sense that the vertex is no longer reflex in either of the two pieces of the partition determined by the cut. Clearly a cut does not introduce any reflex vertices. The induction step of the proof cuts the polygon in two, and applies the induction hypothesis to each half. This will yield the formula of the theorem if a cut can be found such that at least one of the halves contains an odd number of reflex vertices. The only difficult part of the proof is establishing that such an odd-cut always exists. This sketch will now be formalized.

8. An earlier version of this section appeared in O'Rourke (1983b), © 1983 Birkhäuser Verlag.



**Fig. 2.45.** A cut partitions a polygon into two pieces of  $L$  and  $R$  reflex vertices; since the cut resolves one reflex vertex,  $r = L + R + 1$ .

### ***Proof of the Theorem***

The theorem is clearly true for  $r \leq 1$ : a single guard suffices. So assume that  $r \geq 2$  and that the theorem holds for all  $r' < r$ . Consider now two cases.

*Case 1* (There exists a cut that resolves two reflex vertices). This case occurs when two reflex vertices can “see” one another along a vertical or horizontal line. Cut the polygon in two along this line, and let  $L$  and  $R$  be the number of reflex vertices in the two pieces produced. Since  $r = L + R + 2$ , the formula to be proved is

$$\lfloor r/2 \rfloor + 1 = \lfloor (L + R + 2)/2 \rfloor + 1 \geq \lfloor L/2 \rfloor + \lfloor R/2 \rfloor + 2.$$

Applying the induction hypothesis to each half yields a coverage of both polygons (and so the entire original polygon) with  $\lfloor L/2 \rfloor + 1 + \lfloor R/2 \rfloor + 1$  guards, which, by the above calculation, is less than or equal to the formula to be established.

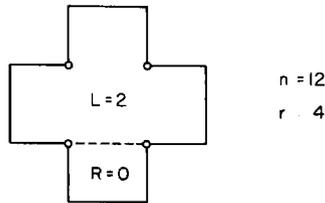
*Case 2* (No two reflex vertices can see one another along a vertical or horizontal line). Lemma 2.17 below will establish that in this case, there exists an *odd-cut*: a cut such that one of the two pieces has an odd number of reflex vertices. Let such a cut be chosen, and let  $L$  and  $R$  be the number of reflex vertices in the halves, with  $R$  odd. As one reflex guard is resolved by the cut,  $r = L + R + 1$ . The formula of the theorem can therefore be written as

$$\lfloor r/2 \rfloor + 1 = \lfloor (L + R + 1)/2 \rfloor + 1 \geq \lfloor L/2 \rfloor + \lfloor (R - 1)/2 \rfloor + 2.$$

Applying the induction hypothesis to each half yields coverage by  $\lfloor L/2 \rfloor + 1 + \lfloor (R - 1)/2 \rfloor + 1$  (since  $R$  is odd), which, by the above calculation, is less than or equal to the formula to be proved.  $\square$

### **2.5.2. Existence of Odd-Cuts**

The existence of odd-cuts will now be established. First note that an odd-cut may not exist if reflex vertices can see one another along horizontal or vertical lines (see Fig. 2.46), but that this falls under Case 1 of the proof

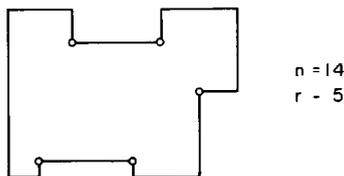


**Fig. 2.46.** A polygon that does not admit an odd-cut, but that permits a single cut to resolve two reflex vertices.

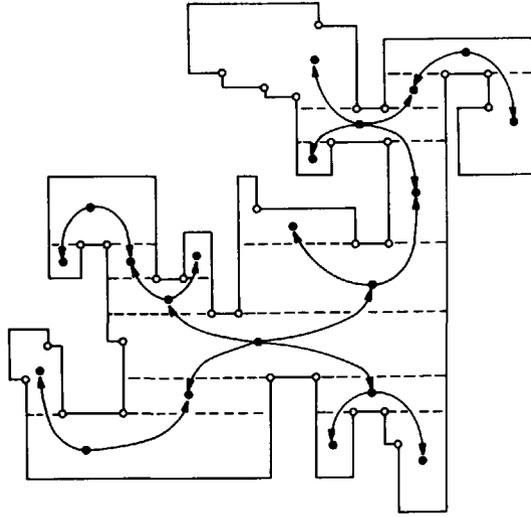
above. Therefore, in this section it will be assumed that the vertices of the polygon are in “general position” in the sense that no cut can resolve two reflex vertices. Second note that the existence of an odd-cut is trivial if  $r$ , the total number of reflex vertices, is even: any cut partitions the reflex vertices according to  $r = L + R + 1$ , so one of  $L$  or  $R$  must be odd and the other even. When  $r$  is odd, either  $L$  and  $R$  are both even or both odd; the task is to show that a cut can be found such that they are both odd. Finally note that a horizontal odd-cut does not always exist: Fig. 2.47 shows an example with  $r = 5$ . In this case, only a vertical odd-cut exists. Thus cuts in both directions must be considered; call a horizontal cut an *H-cut* and a vertical cut a *V-cut*.

The proof depends on a particular orthogonal partitioning of the polygon, which will now be defined for H-cuts. Call a reflex vertex *H-isolated* if the other endpoint of its incident horizontal edge is not reflex, and otherwise call it a member of an *H-pair*. Partition the polygon by forming an H-cut at each reflex vertex that is a member of an H-pair (see Fig. 2.48); the only reflex vertices not resolved in this partitioning are H-isolated vertices. This decomposition is a partition into pieces monotone with respect to the  $y$  axis, as was used in Section 2.3. It will be proved that either an H-odd-cut exists or there is precisely one H-isolated vertex. The proof depends on a rather close analysis of the structure of this partition, which will be explored in terms of its region adjacency graph, called its *H-graph*.

Each piece of the partition corresponds to a node of this graph, and node  $A$  is connected by an arc directed to node  $B$  iff (1)  $A$  and  $B$  are adjacent pieces, separated by an H-cut, and (2) the H-pair corresponding to the H-cut lies on the boundary of the  $A$  piece. See Fig. 2.48. The following lemma classifies the nodes according to their incident arcs.



**Fig. 2.47.** A polygon that has no horizontal odd-cut.



**Fig. 2.48.** The H-graph associated with a polygon records region adjacency in the partitioned formed by cutting at each H-pair member.

**LEMMA 2.13.** The H-graph corresponding to the above defined partition of an orthogonal polygon can have just four types of nodes (see Table 2.2).

**Table 2.2**

Name	Total Degree	Incoming Arcs	Outgoing Arcs
leaf	1	1	0
branch	3	1	2
source	2 or 4	0	2 or 4
sink	2	2	0

*Proof.* The general position assumption prevents a single cut linking two reflex vertices. Thus each region can have at most two H-pairs and therefore four H-cuts on its boundary. Thus the degree of a node is less than or equal to 4. The definition of “arc” implies that a node cannot have just one outgoing arc. Thus a degree 1 node must be a leaf. A degree 2 node can have two outgoing (source) or two incoming (sink) arcs; one outgoing arc is not possible. A degree 3 node must be a branch, again because one outgoing is not possible. A degree 4 node must have two H-pairs on its boundary, which implies that all four arcs are outgoing.  $\square$

It will now be shown that the graph for a polygon that does *not* admit an H-odd-cut must have a very special structure.

**LEMMA 2.14.** If a polygon’s H-graph contains a sink node, then it admits an H-odd-cut.

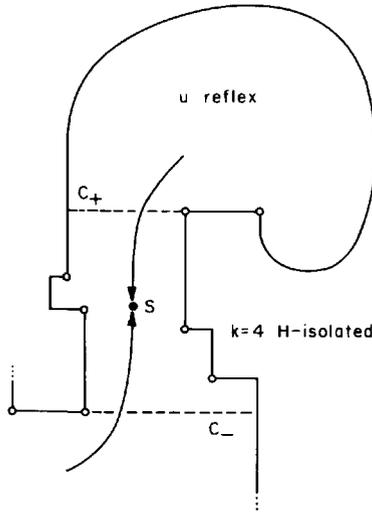


Fig. 2.49. A sink region  $S$  always permits an odd-cut.

*Proof.* Let  $S$  be the region corresponding to a sink node, and let  $C_+$  and  $C_-$  be the upper and lower H-cuts on the boundary of  $S$ . Let  $S$  contain  $k$  H-isolated vertices, and let the total number of reflex vertices in the portion of the original polygon above  $C_+$  (not including the vertex forming  $C_+$ ) be  $u$ . See Fig. 2.49.

If  $u$  is odd, then  $C_+$  is an H-odd-cut. If  $u$  is even, a cut at the highest H-isolated vertex in  $S$  (if  $k > 0$ ) or  $C_-$  (if  $k = 0$ ) is an H-odd-cut.  $\square$

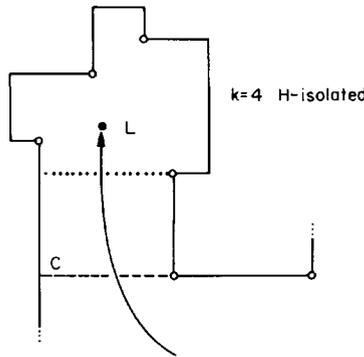
Thus, if a polygon does not admit an H-odd-cut, it cannot have any sink nodes. This implies that such a graph has just a single source node, as two source nodes can only interlink via sinks. Thus the graph is a tree with a source root node, and otherwise binary directed towards the leaves.

**LEMMA 2.15.** If a polygon does not admit an H-odd-cut, then it has exactly one H-isolated vertex located in its sole source region.

*Proof.* First it will be shown that all leaf and branch nodes must be devoid of H-isolated vertices. The proof is by induction on the number of arcs to the nearest leaf node, which we call the *frontier distance*.

Suppose some leaf  $L$  contains  $k > 0$  H-isolated reflex vertices. Let  $C$  be the H-cut corresponding to its single incoming arc (see Fig. 2.50). Then if  $k$  is odd,  $C$  is an odd-cut, and if  $k$  is even, then the H-isolated vertex in  $L$  closest to  $C$  is an odd-cut. This establishes the basis of the induction.

Suppose now that all leaf and branch nodes with frontier distance  $d' < d$  have no H-isolated vertices, and consider a branch node  $B$  at distance  $d$ . Let  $C$  be the H-cut corresponding to its single incoming arc, and let  $k > 0$  be the number of H-isolated vertices in  $B$ . If  $k$  is odd, then  $C$  is an odd-cut, since by the induction hypothesis none of the descendants of  $B$  have any



**Fig. 2.50.** If a leaf region  $L$  has  $k > 0$  H-isolated vertices, then it admits an H-odd-cut.

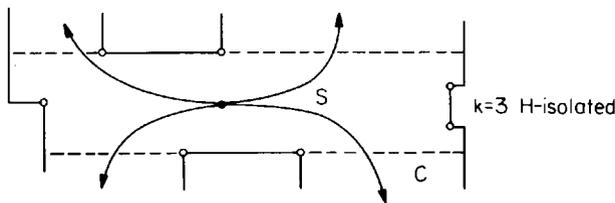
H-isolated vertices, and otherwise the reflex vertices come in H-pairs. If  $k$  is even, then the H-isolated vertex in  $B$  closest to  $C$  can form an odd-cut.

Finally, it will be shown that the single source region  $S$  must have exactly one H-isolated vertex to avoid an H-odd-cut. Suppose  $S$  contains an even number  $k$  of H-isolated vertices, and let  $C$  be a cut corresponding to one of  $S$ 's outgoing arcs (see Fig. 2.51). Then  $C$  is an odd-cut, since it resolves one reflex vertex of an H-pair, and otherwise all other reflex vertices are either in the even  $k$  H-isolated vertices or they come in H-pairs. If  $k$  is odd and greater than 1, then the second closest one to  $C$  forms an odd-cut. Thus there must be exactly one H-isolated vertex.  $\square$

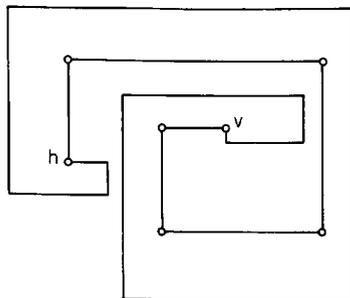
Clearly Lemmas 2.13–2.15 holds for V-cuts as well as H-cuts. Thus, if a polygon does not admit an H-odd-cut nor a V-odd-cut, then it must have a single H-isolated vertex  $h$  and a single V-isolated vertex  $v$ , both located in source regions of the H- and V-graphs, respectively. That these conditions are impossible to achieve is shown by the following lemma.

**LEMMA 2.16.** A polygon of  $r \geq 3$  reflex vertices cannot have exactly one H-isolated vertex in a region corresponding to a source node of its H-graph, and exactly one V-isolated vertex in a region corresponding to a source node of its V-graph.

*Proof.* Let  $h$  and  $v$  be the H- and V-isolated vertices, respectively. All reflex vertices besides  $h$  and  $v$  (there is at least one such since  $r \geq 3$ ) are



**Fig. 2.51.** A source region  $S$  admits an H-odd-cut if it contains  $k \neq 1$  H-isolated vertices.



**Fig. 2.52.** Reflex vertices that are members of both H- and V-pairs form a spiral chain whose end-points are either H- or V-isolated.

members of both an H-pair and a V-pair, else they would be isolated. This implies that they are all adjacent, forming a contiguous chain of reflex vertices. This chain cannot close upon itself without forming a hole, contradicting the assumption that the polygon has no holes. The polygon is therefore a *spiral* (having a single concave chain) whose endpoints are  $h$  and  $v$  (see Fig. 2.52). But then  $h$  is in a leaf region of the H-graph and  $v$  in a leaf region of the V-graph, contradicting the requirement that they be located in source regions.  $\square$

The existence of odd-cuts is now established:

**LEMMA 2.17.** An orthogonal polygon with an odd number  $r \geq 3$  of reflex vertices, no two of which can see one another along a vertical or horizontal line, admits an odd-cut.

This completes the proof of Theorem 2.5.<sup>9</sup> This proof easily extends to “multi-level” orthogonal polygons connected by ramps, as used in Theorem 2.1, but it does not immediately extend to orthogonal polygons with holes. This latter case will be considered in Chapter 5.

## 2.6. ALGORITHM TO PARTITION INTO L-SHAPED PIECES<sup>10</sup>

The proof described in the previous section does not translate directly into an algorithm, because of the difficulty of finding odd-cuts. Nevertheless, we now show that, with the addition of a few new ideas, it can be implemented to run in  $O(n \log n)$  time. Application of the new trapezoidalization algorithm reduces the time to  $O(n \log \log n)$ . We will assume throughout that the polygon is in general position; this is the hard case, as was pointed out in Section 2.5.2.

9. A somewhat simpler proof along similar lines was offered in Manilla and Wood (1984). As this book was being revised a third, even simpler proof was published (Györi, 1986).

10. An earlier version of this section appeared in Edelsbrunner *et al.* (1984), © 1984 Academic Press.

At a coarse level, the algorithm is very simple. First the polygon is preprocessed to detect which horizontal cuts are odd-cuts. Then it is partitioned at every such horizontal odd-cut. Finally, guards are placed in each of the resulting pieces. This will achieve coverage with  $\lfloor r/2 \rfloor + 1$  guards.

The simplicity of this procedure results from care applied at two critical junctures. First, it is easier algorithmically to make an odd-cut in a polygon with an odd number  $r$  of reflex vertices than in one with an even number. The reason, which will be detailed below, is that no updating of the preprocessed computations are necessary for each cut when the number of reflex vertices is odd. We therefore take the counterintuitive step of complicating the polygon by introducing a new reflex vertex if  $r$  is even; the bound of  $\lfloor r/2 \rfloor + 1$  is clearly unaffected.

The second critical juncture arises when all horizontal odd-cuts have been made, and only vertical odd-cuts remain. As previously mentioned, the polygon must then have a restricted structure, and guard placement is nearly trivial. This is fortunate, as otherwise the preprocessing step might have to be repeated with each oscillation between horizontal and vertical cuts.

These claims are justified in the following section.

### 2.6.1. The Algorithm

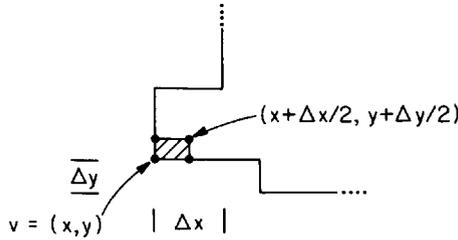
The algorithm for locating  $\lfloor r/2 \rfloor + 1$  guards in an orthogonal polygon of  $n$  vertices,  $r$  of which are reflex, consists of six distinct steps:

- (1) If  $r$  is even, then add an additional reflex vertex.
- (2) Perform a plane sweep to find all horizontal cuts.
- (3) Traverse the boundary once, labeling the parity of the cuts.
- (4) Partition the polygon at each horizontal odd-cut.
- (5) For each resulting piece, place a guard at every other reflex vertex.
- (6) Remove the extra reflex vertex, if introduced in step (1).

Each of these steps will now be described in detail and justified.

#### *Add Reflex Vertex: $O(n)$*

If  $r$  is even, then  $\lfloor (r+1)/2 \rfloor + 1 = \lfloor r/2 \rfloor + 1$ , so the addition of a reflex vertex is justified. The reason for doing so was alluded to above and will be expanded on below. The extra reflex vertex can be added in linear time as follows. Choose an arbitrary convex vertex  $v = (x, y)$ . Find the smallest non-zero horizontal and vertical separations  $\Delta x$  and  $\Delta y$  from  $x$  and  $y$  to other vertices by examining the coordinates of each of the  $O(n)$  vertices. Delete vertex  $v$  and replace it with three others as illustrated in Fig. 2.53. Since no edges of the polygon can cross the rectangle with corners  $(x, y)$  and  $(x + \Delta x/2, y + \Delta y/2)$ , clearly this “dent” maintains the simplicity of the polygon. That it does not interfere with visibility will be shown later.

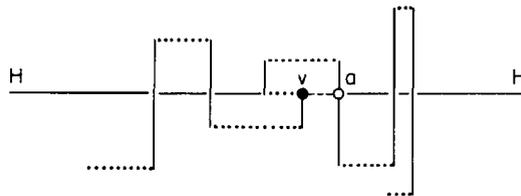


**Fig. 2.53.** A reflex vertex introduced by removing a “chip” from the polygon at a convex vertex  $v$ .

**Plane Sweep for Horizontal Cuts:  $O(n \log n)$  or  $O(n \log \log n)$**

Each reflex vertex determines a unique horizontal cut. The goal of this step is to find each horizontal cut and to insert a new “artificial” vertex into the circular list of vertices representing the polygon at the end of each cut, doubly linking each reflex vertex with its associated artificial vertex. This data structure then serves as input to step (3) of the algorithm.

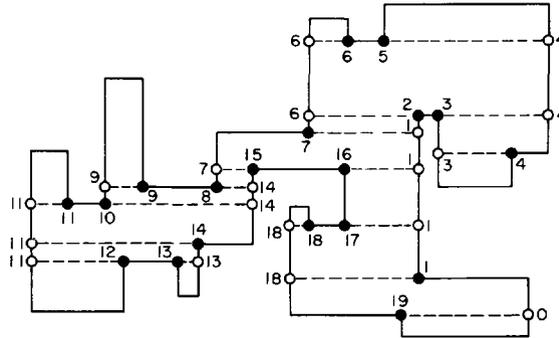
The locations of the artificial vertices are found by a sweep of a horizontal line from top to bottom over the polygon. This is a standard plane sweep, and is nearly identical to the monotone partitioning algorithm of Lee and Preparata (Algorithm 1.2), or to that used in Sections 2.3 and 2.4. The vertical edges of the polygon are sorted by their maximum  $y$  coordinate. At each position of the sweep line  $H$ , a data structure  $S$  holds all those vertical edges pierced by  $H$ , organized left to right. When  $H$  moves down and encounters a reflex vertex  $v$ , the vertical edge hit by  $v$ ’s horizontal cut is available in  $S$  as adjacent to the vertical edge whose top or bottom is  $v$  (see Fig. 2.54). After computing the coordinates of the corresponding artificial vertex, a vertical edge is either inserted or deleted from the data structure  $S$ , depending on whether  $v$  is the top or the bottom of a vertical edge. The data structure can be chosen to support  $O(\log n)$  insertion and deletion time, which leads to  $O(n \log n)$  time overall. Since these horizontal cuts are precisely what the trapezoidalization algorithm of Tarjan and Van Wyk constructs (Section 1.3.2), use of that algorithm instead of a plane sweep reduces the time complexity of this step to  $O(n \log \log n)$ . As this is the only step that might be supra-linear, the entire algorithm is  $O(n \log \log n)$ .



**Fig. 2.54.** When the sweep line  $H$  hits a reflex vertex  $v$ , the artificial vertex  $a$  on its cut lies on a vertical edge adjacent to  $v$ ’s edge in  $S$ .

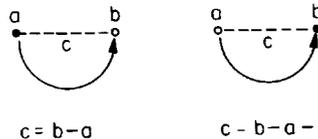
**Boundary Traversal to Compute Cut Parity:  $O(n)$**

The next step is to determine which of the horizontal cuts are odd-cuts—that is, have an odd number of reflex vertices to one side or another. As the total number of reflex vertices  $r$  is known, finding the number to one side of a cut determines the number to the other side. The number to one side of each cut can be found in a single boundary traversal of the polygon as follows.



**Fig. 2.55.** Labels on reflex (solid dots) and artificial vertices (open circles) are generated in a counterclockwise scan of the boundary, incrementing the label counter at each reflex vertex.

Distinguish three types of vertices: convex, reflex, and artificial. Start at an arbitrary vertex, initialize a counter to zero, and proceed counterclockwise around the polygon. If the next vertex is convex, do nothing; if the next vertex is artificial, label it with the counter value; if the next vertex is reflex, increment the counter value and label the vertex with this new value (see Fig. 2.55). As soon as both end points of a cut are labeled, the number of reflex vertices  $k$  to one side is determined by the difference between the two labels. When the artificial vertex of a cut is encountered second, then the exact difference of the labels is  $k$ ; when the reflex vertex is second, then  $k$  is the difference less 1 (see Fig. 2.56).

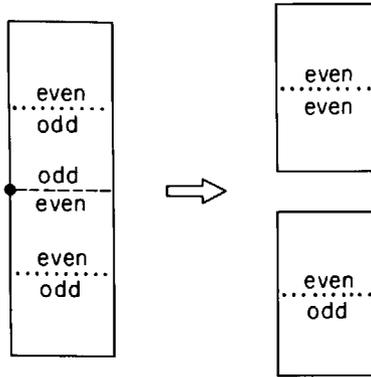


**Fig. 2.56.** The labels  $a$  and  $b$  assigned to the two ends of a cut determine the number of reflex vertices  $c$  to one side by their difference or difference less 1.

It is not actually necessary to compute the *number* of reflex vertices to each side, as only the *parity* is needed. Thus each reflex and artificial vertex can be labeled as even or odd during the traversal, and the parity of the cuts determined by straightforward modification of the rules above.

**Cut at Each Horizontal Odd-Cut:  $O(n)$**

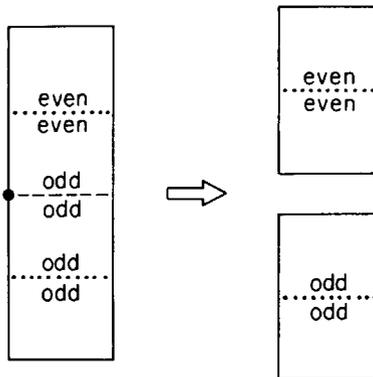
The goal of the fourth step is to cut the polygon at each horizontal odd-cut; the remaining pieces will then be easy to cover with guards. The only



**Fig. 2.57.** Partitioning at an odd-cut can either flip or leave the parity of other cuts unchanged when the total number of reflex vertices is even.

potential difficulty with this step is that, after a particular cut is made, the parities computed in the previous step may no longer be correct for the cuts in the two pieces. For example, Fig. 2.57 shows that a particular odd-cut can either leave the parity of other cuts unchanged, or it could flip their parity, depending on whether the odd number of reflex vertices is inside the portion including the cut or not. However, note that the situations in Fig. 2.57 can only arise when the total number of reflex vertices  $r$  is even. When  $r$  is odd, then there is only one type of odd-cut: a cut that has an odd number of reflex vertices to *each* side. Partitioning the polygon at such an odd-cut leaves the parity of the cuts in each half unchanged (see Fig. 2.58).

Since step (1) of the algorithm guarantees that  $r$  is odd, all the horizontal odd-cuts may be made without any updating to the parities computed in step (3). At the conclusion of this partitioning, we are left with several orthogonal polygons, all of which have only even horizontal cuts remaining.



**Fig. 2.58.** If the total number of reflex vertices is odd, then partitioning at an odd-cut does not affect the parity of other cuts.

**Guard Placement:**  $O(n)$ 

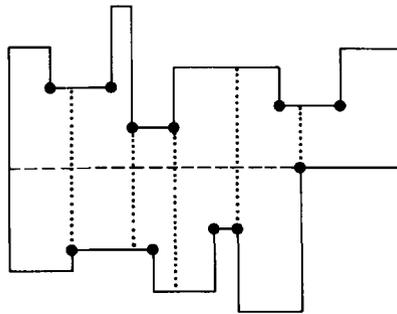
Consider now an orthogonal polygon with no horizontal odd-cuts. If it has no reflex vertices, then it is a rectangle and can be covered with a single guard placed anywhere within the interior. This satisfies the bound  $\lfloor r/2 \rfloor + 1$  of the theorem.

If there is at least one reflex vertex, then Lemma 2.17 establishes that a vertical odd-cut exists. However, finding such a cut would require repetition of the previous four steps of the algorithm, and ultimately may lead to switching back to horizontal cuts again. Such oscillation and the computation it entails could be very expensive. Fortunately, polygons without horizontal odd-cuts have a special structure that makes guard placement easy.

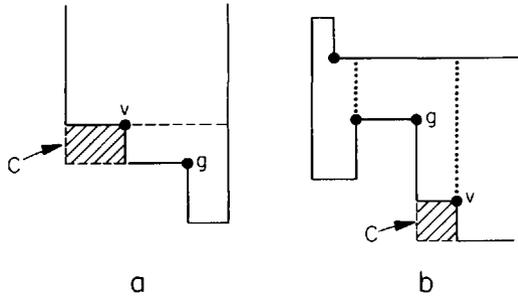
It was proven in Section 2.5.2 that a polygon with only even horizontal cuts must appear as in Fig. 2.59: it is formed by two “histograms” joined at their bases. This structural restriction was stated in a different but equivalent terms in Lemma 2.15. More precisely, define a *vertical histogram* as in Section 2.3.2: an orthogonal polygon that has one horizontal edge (the base) equal in length to the sum of lengths of all the other horizontal edges. Then a polygon that has no horizontal odd-cuts must have a reflex vertex whose horizontal cut partitions the polygon into two vertical histograms (as in Fig. 2.59). Such a polygon clearly must have an odd number of reflex vertices.

Guards placed at every other reflex vertex from left to right will necessarily cover such a polygon. To see this, let the reflex vertices be  $v_1, v_2, \dots, v_k$  in sorted order from lowest  $x$  coordinate to highest; here  $k$  is odd. Make a vertical cut at  $v_2, v_4, \dots, v_{2\lfloor k/2 \rfloor}$ . The polygon has now been partitioned into  $\lfloor k/2 \rfloor + 1$  L-shaped pieces, each containing one reflex vertex (see Fig. 2.59). Place guards on  $v_1, v_3, \dots, v_{2\lfloor k/2 \rfloor + 1}$ . This clearly covers the polygon with  $\lfloor k/2 \rfloor + 1$  guards, achieving the bound of the theorem.

The sorting of the reflex vertices can be accomplished in linear time by



**Fig. 2.59.** Polygons with no horizontal odd-cuts have a horizontal cut (dashed) that partitions them into two vertical histograms, which can then be cut vertically into L-shaped pieces at every other reflex vertex.



**Fig. 2.60.** The chip region  $C$  is visible to some guard  $g$ , regardless of whether a horizontal (a) or vertical (b) cut emanates from  $v$ .

merging the two histogram chains, which are necessarily already sorted. Once the order is determined, no vertical cuts need actually be made: the guards are simply assigned to every other reflex vertex. Note that after step (5) is completed, the polygon has been implicitly partitioned into L-shaped pieces; one of these may be modified to a rectangle in the final step discussed next.

Finally, we note that the same simple guard location procedure would work if the polygon were monotone with respect to the  $x$  axis. This is a wider class of shapes than the double histograms that result from step (5).

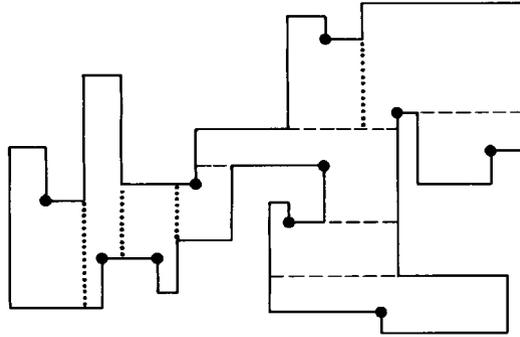
### **Replacement of Chip: $O(n)$**

If an extra reflex vertex was introduced by removing a “chip”  $C$  at a convex corner of the polygon in step (1) of the algorithm (see Fig. 2.60), then returning the polygon to its original form will not require any more guards, nor even a movement of the existing guard locations.

Consider two cases. If the introduced reflex vertex  $v$  is assigned a guard during step (5), then retract the chip back to the original convex corner, and place the guard in that corner. Clearly the L-shaped region formerly covered by the guard at  $v$  is now a rectangle and covered by the guard in the corner. If  $v$  is not assigned a guard, then a cut was made at that vertex in the algorithm, either a horizontal cut in step (3) or a vertical cut (implicitly) in step (4). In either case, one of the two edges incident to  $v$  is a complete edge of an L-shaped region that has a guard  $g$  in it (see Fig. 2.60). This follows since the edges of the chip were chosen too small to be hit by any cuts. Clearly the guard  $g$  sees into  $C$ .

### **2.6.2. Discussion**

We have assumed that no two reflex vertices of the polygon can see one another along a vertical or horizontal line. Lemma 2.17 depends on this assumption: without it a polygon may have no odd-cuts, either horizontal or vertical (Fig. 2.46). Fortunately this degeneracy is in our favor, so to speak, and is easy to handle.



**Fig. 2.61.** Guards positioned in the polygon of Fig. 2.55. The dashed lines are the horizontal cuts made in step (4) of the algorithm, and the dotted lines are the vertical cuts implicit in step (5). Here  $r = 19$ , and  $\lfloor r/2 \rfloor + 1 = 10$  guards are used.

If two reflex vertices see one another along a horizontal, this can be detected during the plane sweep, step (2) of the algorithm. Before commencing step (3), we can cut the polygon into pieces at each such horizontal. As established in Section 2.5.1, recursive application will achieve the desired bound, and each piece can be processed by the algorithm separately.

If two reflex vertices see one another along a vertical, one of the L-shaped regions formed by the implicit vertical cuts in step (5) may degenerate to a rectangle, but this in no way affects the execution of the analysis of the algorithm.

The guard locations chosen by the algorithm for the example used in Fig. 2.55 and the partition they induce, are shown in Fig. 2.61.<sup>11</sup>

<sup>11</sup> I thank Carmen Castells and David Shallcross for implementing this algorithm.