DECOMPOSING POLYGONAL REGIONS INTO CONVEX QUADRILATERALS

Anna Lubiw

Department of Computer Science
University of Toronto
Toronto, Canada, M5S 1A4

1. Introduction

A ***polygonal region*** is a closed region of the plane formed by cutting holes bounded by polygons out of a region bounded by a polygon. ***Vertices*** [***edges***] of a polygonal region are vertices [edges] of the bounding polygon and the hole polygons. A ***chord*** of a polygonal region is a line segment inside the region joining two vertices. ***Decomposing a polygonal region into quadrilaterals*** means adding chords, no two of which cross, so that the minimal regions formed by the chords and edges are quadrilaterals. This work is about decomposing polygonal regions into convex quadrilaterals.

The problem of deciding whether a polygonal region can be decomposed into convex quadrilaterals is shown to be **NP**-complete. Polygonal regions without holes are more tractable: section 7 contains a polynomial-time dynamic programming algorithm to find--given a polygonal region without holes, and given weights on the convex quadrilaterals formed by edges and chords of the region--a decomposition of the region into convex quadrilaterals which minimizes the sum of the weights of the quadrilaterals of the decomposition.

A main purpose of this work is to characterize some classes of polygonal regions which can be decomposed into convex quadrilaterals. In [KKK] Kahn, Klawe, and Kleitman proved that rectilinear regions can be decomposed into convex quadrilaterals. A ***rectilinear polygon*** is a polygon whose every side is horizontal or vertical; and a ***rectilinear region*** is a polygonal region whose bounding polygon and hole polygons are all rectilinear.

Supported by NSERC

I take a new approach to the [KKK] result, by showing that two larger classes of polygonal regions P_1 and P_2 are CQ hereditary. A class P of polygonal regions is ***CQ hereditary*** if every polygonal region in P which is not itself a convex quadrilateral has a ***removable quadrilateral***, defined to be a convex quadrilateral contained in the region whose sides are edges or chords of the region and whose removal leaves polygonal regions in the class. Obviously any polygonal region in a CQ hereditary class can be decomposed into convex quadrilaterals.

The class P_1 contains rectilinear regions without holes, and the class P_2 contains rectilinear regions even with holes. It is an open problem to find a natural unification of these two classes. The proof that P_2 is CQ hereditary can be made into an $O(n \log n)$ algorithm to decompose n-vertex polygonal regions of P_2 into convex quadrilaterals. In particular this algorithm works for rectilinear regions, and for these I prove that $O(n \log n)$ is best possible in a fairly general computation model.

Figure 1: a polygonal region not decomposable into convex quadrilaterals.

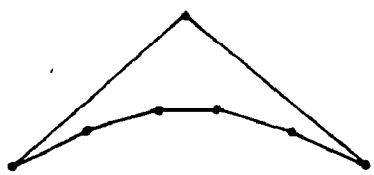
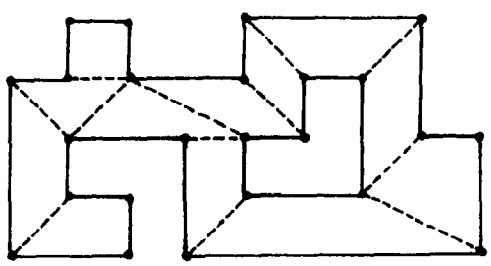


Figure 2: a rectilinear region decomposed into convex quadrilaterals.



2. Background

An alternative concept of decomposition is that of "Steiner decomposition" in which new vertices may be added to the interior or boundary of the polygonal region before chords are added. Decomposing polygonal regions into simpler polygonal regions is an important problem in computational geometry with applications in computer graphics, pattern recognition, and VLSI design (see [T], which is a survey of computational geometry). "Simpler polygonal regions" may be simpler by virtue of having few vertices (triangles, quadrilaterals, etc.) or by having some special structure (convex, monotone, star-shaped, etc.), or by a combination of these things.

Results related to the present one are: Sack [S] claims an $O(n\log n)$ algorithm, using the [KKK] proof, to decompose n-vertex rectilinear regions without holes into convex quadrilaterals. [GJPT] contains an $O(n\log n)$ algorithm to decompose an n-vertex polygonal region without holes into triangles. Keil [K] has given a polynomial time algorithm to decompose a polygonal region without holes into a minimum number of convex polygons, and shown that the problem is **NP**-hard if holes are allowed.

Related results for Steiner decompositions are: [PLLML] contains a polynomial time algorithm to find a Steiner decomposition of a rectilinear region into a minimum number of rectangles. [LPRS] contains a polynomial-time algorithm to find a Steiner decomposition of a rectilinear region without holes into rectangles minimizing the sum of the lengths of the line segments added, and a proof that this problem is **NP**-hard when holes are allowed. [CD] gives a polynomial time algorithm to find a Steiner decomposition of a polygonal region without holes into a minimum number of convex polygons, and [Ln] shows that this problem in **NP**-hard if holes are allowed.

The motivation in [KKK] for proving that any rectilinear region can be decomposed into convex quadrilaterals is to prove that all points of an n-vertex rectilinear region without holes can be seen from a set of at most $\lfloor n/4 \rfloor$ vertices. (Or, more colloquially, $\lfloor n/4 \rfloor$ watchmen can guard a conventional art gallery with n walls.) The argument (which follows [F]) is elegant: After adding chords to decompose the rectilinear region into convex quadrilaterals, colour the vertices with four colours so that each quadrilateral of the decomposition gets all four colours. This is possible if the region has no holes. The smallest colour class has at most $\lfloor n/4 \rfloor$ vertices and can see the whole region.

O'Rourke in [O] has given an alternative proof of this watchmen result which avoids the intermediate result on decompositions into convex quadrilaterals. And [EOW] contains an $O(n\log n)$ algorithm to position a set of at most $\lfloor n/4 \rfloor$ watchmen to guard an n-vertex rectilinear polygon.

3. Groundrules

No edge of a polygonal region may *cross* another edge--i.e. intersect the edge and contain points on either side of it. But apart from this, edges will be allowed to overlap.

Consecutive collinear vertices count. Thus the polygonal region in Figure 3 can be decomposed into convex quadrilaterals, but the one in Figure 4 cannot.

Figure 3.

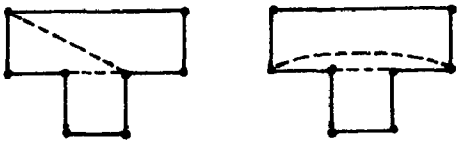
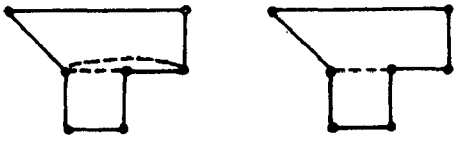


Figure 4.



More generally:

Proposition 0. If an n-vertex polygonal region with h holes can be decomposed into q quadrilaterals then $q = \frac{1}{2}(n+2h-2)$. In particular n is even.

4. P_1: 1-Rectilinear Regions

A *1-rectilinear region* is a polygonal region without holes and with a distinguished edge e called the *tilted edge* such that

(1.1) there are an even number of edges,

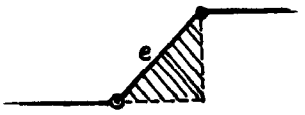
(1.2) the edges except possibly e are alternately (around the polygon) horizontal and vertical,

(1.3) all interior angles are $\leq 270°$,

(1.4) the nose of the tilted edge contains no vertices.

The *nose* of the tilted edge is the triangular region with one horizontal side, one vertical side, and the tilted edge as hypotenuse--and attached to the inside of the tilted edge. The nose is closed along the hypotenuse, open along the other two sides, and excludes the three corners. The nose of a horizontal or vertical edge is empty.

Figure 5: the nose of e



A rectilinear region without holes is 1-rectilinear since any edge can be the tilted one.

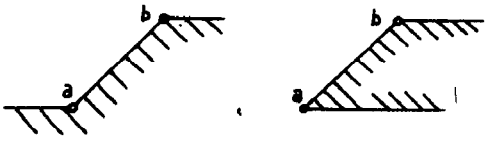
Theorem 1. The class of 1-rectilinear regions is CQ hereditary.

A vertical edge of a 1-rectilinear region is a ***left edge*** if the inside of the region is to the right of the edge, and otherwise it is a ***right edge***. A horizontal edge of a 1-rectilinear region is a ***top edge*** if the inside of the region is below the edge, and otherwise it is a ***bottom edge***.

Proof of Theorem 1.

Finding a removable quadrilateral in a 1-rectilinear region: The two edges incident with the tilted edge $e=(a,b)$ must be both horizontal or both vertical. Assume without loss of generality one of the two cases shown in Figure 6. (Other cases are obtained by rotations and/or reflections.)

Figure 6.



Find the vertex p of minimum x then maximum y in the shaded area shown in Figure 7. This area is closed on the left and top, open on the bottom, and excludes the two corners. Since the region contains at least one vertex--namely the other end of the horizontal edge incident with b--p exists. p is at the top of a right edge and/or at the other end of the horizontal edge incident with b. See Figure 9.

Figure 7.

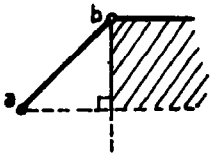
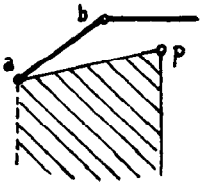


Figure 8.



Look for a vertex q of maximum y then maximum x in the shaded area shown in Figure 8. This area is closed on the top and right, open on the left, and excludes a and p. If vertex p is at the top of a vertical edge then q exists and is at the right of a bottom edge and/or at the other end of the vertical edge incident with p. If vertex p is not at the top of a vertical edge then it may happen that q does not exist, or that q exists but is at the left end of a bottom edge. Then (see Figure 10) the vertical line extending down from p must meet a bottom edge f. Note that f cannot be higher than a. In this case find the vertex q of minimum x then minimum y in the shaded area shown in Figure 10. This area is closed on the left and bottom, open on the top, and excludes the two corners. q exists (since the right end of the horizontal edge f is a vertex of the region) and is at the bottom of a right edge and/or at the right end of f.

Figure 9: possibilities for p.

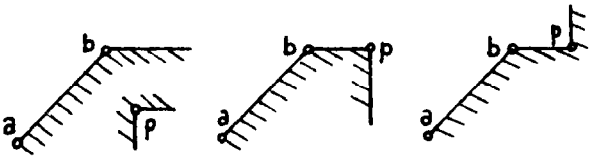
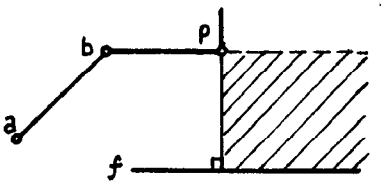


Figure 10.



Proof that the quadrilateral $abpq$ is removable: By choice of p and q, each of the four sides of the quadrilateral is inside the region, and each of the four angles is less than or equal to 180°. Thus $abpq$ is a convex quadrilateral contained in the 1-rectilinear region.

The polygonal regions formed by removing quadrilateral $abpq$ clearly satisfy conditions (1.2) and (1.3). Vertices of the 1-rectilinear region can be divided into: type 1--vertices at the top of a right edge, the bottom of a left edge, the right of a top edge, or the left of a bottom edge; and type 2--vertices at the bottom of a right edge, the top of a left edge, the left of a top edge, or the right of a bottom edge. Vertices are alternately of type 1 and type 2 around the polygon. Since a is of type 1, b of type 2, p of type 1, and q of type 2, the polygons formed by removing quadrilateral $abpq$ satisfy condition (1.1).

The nose on edge bp in the new polygonal region containing this edge contains no vertices of the polygon by the choice of p. Similarly the noses on edges aq and qp in their polygonal regions contain no vertices by definition of q. Thus the polygonal regions formed by removing quadrilateral $abpq$ satisfy condition (1.4), and are 1-rectilinear ■

This method provides a polynomial time algorithm to decompose 1-rectilinear regions into convex quadrilaterals.

5. P_2: Pseudo-Rectilinear Regions

A *pseudo-rectilinear region* is a polygonal region such that

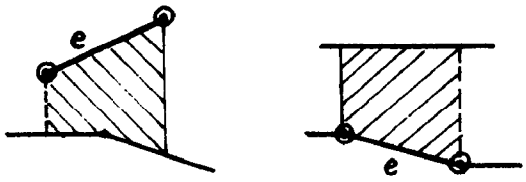
(2.1) alternate edges around all polygons forming the region are horizontal (other edges are called *tilted*),

(2.2) interior angles are $\leq 270°$,

(2.3) the shadow of any tilted edge contains no vertices.

The *shadow* of an edge e of a polygonal region consists of the points inside the region which can be connected to a point on e by a vertical line segment contained in the region--with the exception of the endpoints of e, points vertically below the left endpoint of e, and points vertically above the right endpoint of e.

Figure 11: shadow of e.



Any rectilinear region is pseudo-rectilinear.

Theorem 2. The class of pseudo-rectilinear regions is CQ hereditary.

Denote by $<_x$ the ordering of vertices by x coordinate and when these are equal by y coordinate. Denote by $<_y$ the ordering of vertices by y coordinate and when these are equal by x coordinate. A tilted edge of a pseudo-rectilinear region is a *left edge* if the inside of the region is to the right of the edge travelled from its vertex minimum in $<_y$ to its other vertex; a tilted edge is a *right edge* otherwise. Two vertices *see* each other (are *visible* from each other) if the line segment joining them is contained in the region. The *right neighbour* of vertex v is the vertex (if it exists) minimum in $<_x$ from the set of vertices $r >_x v$ visible from v. The *rightmost vertex* of an edge is its vertex maximum in $<_x$.

Proof of Theorem 2.

Finding a removable quadrilateral in a pseudo-rectilinear region: Find a left edge (u,v), $u <_x v$ where v has a right neighbour r on a right edge (r,s). The pair of edges (u,v), (r,s) form a quadrilateral Q which is removable.

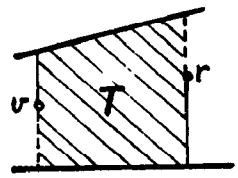
Lemma 3. In a pseudo-rectilinear region every vertex b on a left edge (a,b), $a <_x b$, has a right neighbour.

Then if left edges are ordered by the ordering $<_x$ on right endpoints, the right neighbour of the right endpoint of the rightmost left edge must be on a right edge. Therefore a pair $(u,v),(r,s)$ exists.

Proof of Lemma 3. It suffices to show that b sees a vertex c, $c >_x b$. If the horizontal edge incident with b goes to a vertex of higher x coordinate this vertex provides a c. Otherwise every ray from b with angle $\vartheta \in (-90°, 90°)$ begins inside the region. ($0°$ is the ray in the direction of the $+x$ axis.) Each ray exits the region at some point. If some ray exits at a vertex this vertex provides a c. If two rays exit at different edges of the region, some ray between them must exit at a vertex. But it cannot happen that rays arbitrarily close to both $90°$ and $-90°$ leave the region at the same edge.

Proof that quadrilateral Q is removable: Let T be the union of the two shadows of edge (v,r) in the two polygonal regions formed by cutting along chord (v,r). See Figure 12. Since r is the right neighbour of v, T contains no vertices.

Figure 12.



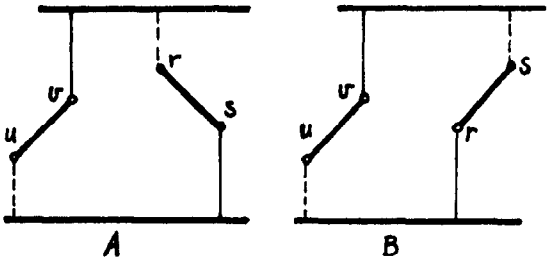
If $s <_x r$ then the right edge (r,s) must bound T on the top or bottom, and v will be in the shadow of (r,s). This proves:

Lemma 4. If vertex v on left edge (u,v), $u <_x v$ has right neighbour r on right edge (r,s) then $u <_x v <_x r <_x s$.

Without loss of generality suppose $u <_y v$. This leaves two cases: $s <_y r$ as in Figure 13A, and in which case Q is (in order) $uvrs$; or $s >_y r$ as in Figure 13B, and in which case Q is $uvsr$.

Let S be the union of T, the shadow of (u,v), and the shadow of (r,s). Note how closed and open boundaries match up. S has no vertices in it. Q is contained in S, and therefore contained in the pseudo-rectilinear region. Once Q is removed (v,r) and (u,s) in case A, and (v,s) and (u,r) in case B will become tilted edges unless they are already horizontal edges. The shadows of these edges are contained in S and so have no vertices in them. Thus the polygonal regions formed by removing Q satisfy (2.3), and since they clearly satisfy (2.1) and (2.2), therefore they are pseudo-rectilinear.

Figure 13.



It remains to be shown that Q is convex. Clearly the angles at u and s are less than or equal to 180°. Since $u <_x v$ and $u <_y v$ the angle of the pseudo-rectilinear region at v can be greater than 180° only if the horizontal edge incident with v heads in the negative x direction. In this case--since the angle at v is bounded by 270°--u must have the same x coordinate as v. But then the angle of Q at v cannot be greater than 180°, because $r,s >_x v$. A similar argument shows that the angle of Q at r is no more than 180° ■

6. An Optimal Algorithm for Decomposing Pseudo-Rectilinear Regions into Convex Quadrilaterals

This section contains an $O(n \log n)$ implementation of the method described in the previous section for decomposing n-vertex pseudo-rectilinear regions into convex quadrilaterals. The machine model is a random access machine or RAM (see [AHU]) with indirect addressing, branching based on comparisons, and the arithmetic operations of $+,-,\times$. Inputs are integers and the uniform cost measure is used. Note that division, and Boolean operations on the binary representations of numbers are not allowed. Multiplications will be used in the algorithm only if the given polygonal region is not rectilinear and should be tested for pseudo-rectilinearity.

For this model of computation I will show that $O(n \log n)$ is a lower bound on the time needed to decompose n-vertex rectiliner regions into convex quadrilaterals. The proof uses the result of [PS] that sorting n numbers requires $O(n \log n)$ time on this restricted RAM model.

Input to the algorithm is a list of polygons forming the polygonal region. Each polygon is given by a list of vertices in clockwise or counterclockwise order around the polygon. Each vertex is given by integer-valued x and y coordinates. It will be assumed that no two edges cross--i.e. that the input specifies a polygonal region. A method of testing this assumption in time $O(n \log n)$ can be found in [SH]. I will describe how to test that the region is pseudo-rectilinear. Note that testing rectilinearity of a polygonal region is easy.

Output from the algorithm is a list of the quadrilaterals used in the decomposition, each quadrilateral specified by four vertices in clockwise order.

No connectivity information will be kept: the algorithm will operate on a disjoint union of pseudo-rectilinear regions. For each vertex v the vertex joined to v by a horizontal edge and the vertex currently joined to v by a tilted edge will always be available.

Consider first the problem of finding a left edge (u,v) such that v's right neighbour is on a right edge. As noted in the previous section, if left edges are ordered by the ordering $<_x$ on right endpoints, the right neighbour of the right endpoint of the rightmost left edge must be on a right edge. Furthermore, when the convex quadrilateral determined by the last left edge is removed that left edge disappears; up to two new left edges may be created; and, by Lemma 4, they go at the end of the ordered list of left edges. An ordered list of left edges will be maintained through the course of the algorithm.

Consider next the problem of finding right neighbours. These can change through the course of the algorithm. Define the *initial right neighbour* of vertex v in a pseudo-rectilinear region to be the vertex (if it exists) minimum in $<_x$ from the set of vertices $r >_x v$ visible from v and not joined to v by a tilted edge.

Lemma 5. Suppose the disjoint union of pseudo-rectilinear regions P' results from the pseudo-rectilinear region P by removing convex quadrilaterals determined by pairs $(u,v),(r,s)$ where (u,v), $u <_x v$ is a left edge, v's right neighbour is r, and (r,s) is a right edge. If v is a rightmost vertex on a left edge in P' then the right neighbour of v in P' is the initial right neighbour of v in P.
(Proof later.)

So if initial right neighbours are calculated for a given pseudo-rectilinear region then the decomposition algorithm need never search for right neighbours.

Algorithm

Step I. Verify that the polygonal region P is pseudo-rectilinear; find initial right neighbours in P; and create a list L of left edges of P ordered by the ordering $<_x$ on rightmost vertices. See details below.

Step II. Repeat until the list L of left edges is empty:

1. Remove the last left edge $(u,v), u <_x v$, from L.
2. Let r be the initial right neighbour of v, and s be the vertex currently joined to r by a tilted edge.

3. Output the quadrilateral formed by (u,v), (r,s) in the appropriate order as follows: Let u'=max wrt $<_y$ of u,v; v'= other of u,v; r'=max wrt $<_y$ of r,s; s'= other of r,s. Output v',u',r',s'.

4. Take care of the new tilted edges as follows: If (u',r') is not a horizontal edge record that it has become a tilted edge. If in addition $u'>_y r'$ then add (u',r') to the end of L. If (v',s') is not a horizontal edge then record that it has become a tilted edge; If in addition $v'<_y s'$ then add (v',s') to the end of L--but put it before (u',r') if (u',r') was added to L and $s'<_x r'$.

Note that Step II can be done in time linear in n on the specified machine model without multiplications.

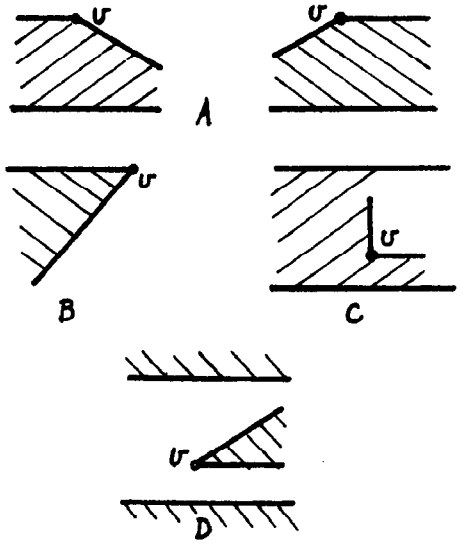
Implementing Step I: A scanning technique will be used. First order the vertices by $<_x$ in $O(n\log n)$ time. A vertical "scan line" will be swept from $x=-\infty$ to $x=\infty$, stopping at each vertex in order. An ordered list of the edges of P crossing the current scan line will be kept. Only the identity of the edge will be kept, not the exact y value at which the edge intersects the scan line; By the assumption that edges do not cross, the relative ordering of any two edges is constant as the scan progresses. The list of edges must be stored in such a way that the operations of searching for an element, adding an element, or deleting an element can each be done in time $O(\log m)$ where m is the length of the list. A balanced search tree (see [AHU] §4.9, or [Ta] §4.2) will do, and can be implemented on the specified machine model.

Edges come in pairs as the scan line from bottom to top successively enters and leaves the region. Note that one edge out of each such "in-out" pair must be horizontal if property (2.3) holds. With each "in-out" pair will be associated a list of vertices awaiting initial right neighbours.

Suppose vertex v is the next vertex to be encountered in the scan. Find its place relative to the ordered list of edges. It may be on one edge (case A), on two edges (case B), between two edges forming an in-out pair (case C), or between two edges of different in-out pairs (case D). See Figure 14.

Testing whether v is an endpoint of an edge is easy; testing whether v is above or below a horizontal edge can be done by comparing y coordinates; testing whether v is above or below a tilted edge requires multiplications of coordinate values as well as comparisons. Note that such tests can be avoided if the region is assumed to satisfy condition (2.3).

Figure 14.



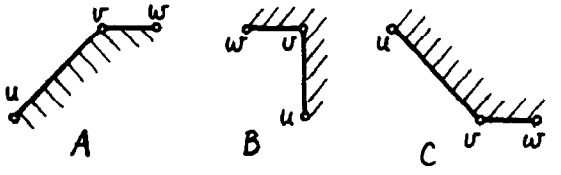
Once v's position relative to the current edge list is known the edge list must be updated appropriately. In cases C and D two new edges must enter the list; in case A one edge is replaced by another; in case B two edges disappear. In cases A, B and C there may be vertices awaiting initial right neighbours associated with the in-out pair which v lies between. Any such vertices not joined to v by a tilted edge should be assigned v as an initial right neighbour. Any such vertices joined to v by a right edge have no initial right neighbour. Any such vertices joined to v by a left edge can continue to await an initial right neighbour. Also, in cases A, C, and D, v itself (associated with the appropriate in-out pair) should now await an initial right neighbour.

Left edges can be identified during the scan. They can later be sorted according to $<_x$ on right endpoints. Testing that alternate edges are horizontal and that no interior angle is greater that 270° can be done during the scan. Testing for the second condition requires only comparisons of coordinates by virtue of the first condition.

The scan takes time $O(n\log n)$ on the specified machine model.

Proof of Lemma 5. Suppose vertex v is on a left edge $(u,v), u<_x v$ in P'. Let (v,w) be the horizontal edge incident with v. Through the course of the algorithm the horizontal edge incident with a vertex remains the same, but the tilted edge may change as the interior angle at the vertex gets smaller. If $u<_y v$ then either $w>_x v$ (case A), or $w<_x v$ and u and v are at the same x coordinate (case B). If $u>_y v$ then $w>_x v$ (case C). See Figure 15. If case A occurs in P' it is possible that in P the tilted edge incident with v went vertically upward from v. Call this case A0. Except in case A0 vertex v will always have been the rightmost vertex of the tilted edge incident with it.

Figure 15.



Let r be the initial right neighbour of v in P. Note that except in case A0, r is the right neighbour of v in P. Let T be the union of the two shadows of edge (v,r) in the two polygonal regions formed from P by cutting along chord (v,r). Since r is the initial right neighbour of v, T contains no vertices. r will be the right neighbour of v in P' unless during the course of the algorithm the removal of some convex quadrilateral Q prevents v from seeing r. Since T contains no vertices, such a Q must contain a vertex $p \leq_x v$ and a vertex $q \geq_x r$. If v is not a vertex of Q then v is in the shadow of one of the edges formed by removing Q, in contradiction to the proof of Theorem 2. On the other hand, suppose v is a vertex of Q. If v is on a right edge forming Q then either case A0 or else v is rightmost on the right edge, but in either case Q cannot contain a vertex $q \geq_x r$. If v is on a left edge forming Q then by Lemma 4 the removal of Q either eliminates v or makes v the leftmost vertex of a tilted edge, contradiction to v being the rightmost vertex of a left edge later on ■

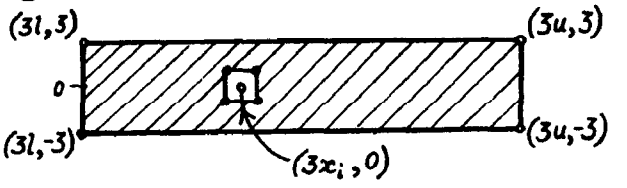
Theorem 6. On a unit cost RAM with indirect addressing, branching based on comparisons, and the arithmetic operations of $+,-,\times$, any algorithm to decompose n-vertex rectilinear regions into convex quadrilaterals requires $O(n \log n)$ time.

Proof. The problem of sorting n integers will be reduced to the problem of decomposing a $(4n+4)$-vertex rectilinear region into convex quadrilaterals. The reduction can be carried out on the specified RAM model in $O(n)$ time. It is proved in [PS] that sorting n integers requires $O(n \log n)$ time on this model. So the theorem is proved.

Given n integers $x'_1, x'_2, \ldots, x'_n$ make them distinct by multiplying them by n and adding i to the i^{th} one: $x_i \leftarrow n \cdot x'_i + i$. Find lower and upper bounds l, u such that $l \leq x_i \leq u$ for all $i = 1, \ldots, n$. Make a rectilinear region bounded by the rectangle $(3l,-3)$, $(3l,3)$, $(3u,3)$, $(3u,-3)$, and with, for each i, a square hole of side length 2 centered at $(3x_i,0)$. See Figure 16.

The rightmost side $((3x_i+1,-1),(3x_i+1,1))$ of the hole for x_i can be in a convex quadrilateral only with the leftmost side of the next hole to the right. Thus a decomposition of the rectilinear region into convex quadrilaterals provides a link from each x_i to the smallest x_j greater than x_i. Following these links sorts the x_i's. This reduction can be carried out in $O(n)$ time on the specified model ■

Figure 16.



It is an open question whether $O(n \log n)$ is a lower bound on the time required for decomposing n-vertex rectilinear regions without holes into convex quadrilaterals.

7. Decomposing Polygonal Regions Without Holes into Convex Quadrilaterals

This section contains a polynomial-time dynamic programming algorithm to find--given a polygonal region P without holes, and given weights $w(q)$ on the convex quadrilaterals q formed by chords and edges of the region--a decompostition of the region into convex quadrilaterals which minimizes the sum of the weights of the quadrilaterals used in the decomposition. In particular if the weight of a convex quadrilateral is taken to be the Euclidean length of its perimeter (assuming Euclidean distances can be computed) then the algorithm finds a decomposition into convex quadrilaterals which minimizes the sum of the lengths of the chords used. This is called a "minimum ink" decomposition.

Fix an edge e of the polygonal region P with chords C. Any chord c divides the region into two polygonal regions without holes; let P_c denote the one not containing e. Let $P_e = P$. For $c \in C \cup \{e\}$ let $m(P_c)$ denote the minimum weight sum of a decomposition of P_c into convex quadrilaterals. Then $m(P_k)$ for $k \in C \cup \{e\}$ is the mnimum over convex quadrilaterals q contained in P_k and containing k of $w(q) + \sum\{m(P_c): c \neq k, c$ a chord forming $q\}$. Thus if $C \cup \{e\}$ is partially ordered: $c_1 < c_2$ if c_1 is a chord of P_{c_2}, then as c goes from min to max the recursion formula can be used to find a decomposition of P_c into convex quadrilaterals whose weight sum is $m(P_c)$. At the end this gives a decomposition of $P = P_e$ into convex quadrilaterals whose weight sum is $m(P)$.

8. The General Convex Quadrilateral Decomposition Problem is NP-Complete

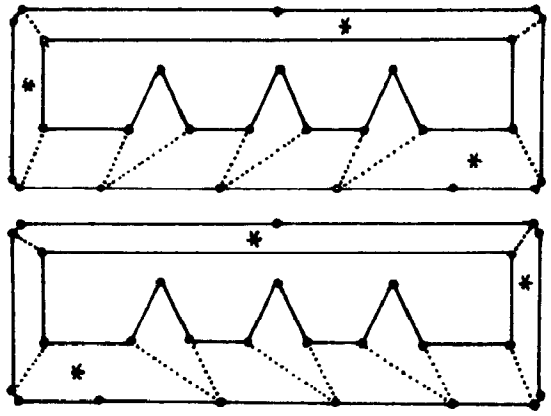
Theorem 7. The problem of deciding whether a polygonal region can be decomposed into convex quadrilaterals is **NP**-complete.

Proof. Clearly the problem is in **NP**. The reduction will be from the PLANAR 3-SATISFIABILITY problem: given a Boolean formula F in conjunctive normal form with variables V and clauses C, each clause a disjunct of 3 literals, and such that the bipartite graph $G=(V\cup C,E)$ where $E=\{(v,c): v\in V, c\in C, v$ or $\bar{v}$ is a literal in $c\}$ is planar, decide if F is satisfiable. PLANAR 3-SATISFIABILITY is **NP**-complete: see [GJ],p. 259 or the original reference [L].

I will show how to construct, from an instance of PLANAR 3-SATISFIABILITY as above, a polygonal region based on a planar embedding of G which is decomposable into convex quadrilaterals if and only if F is satisfiable.

Replace each node of V in the planar embedding of G by a rectangular ring as shown in Figure 17. This structure can be decomposed into convex quadrilaterals in two essentially different ways (as shown) which will correspond to setting the variable true or false. The rectangular ring can be lengthened allowing for more or fewer triangular points.

Figure 17: the rectangular ring for a variable and its two decompositions. Areas marked * can be decomposed in more than one way.



For each edge incident with a node of V in G construct a pair of channels leaving the rectangular ring as shown in Figure 18. Note that either the chord ab or the chord cd must be used. The possible decompositions are shown in Figure 19.

It is not possible for both triangular points (A and B in Figure 18) to use the inner vertices (e and f, respectively). Thus the case where both triangular points use the outer vertices (Figure 19(c)) is ruled out because it would mess things up somewhere else on the rectangular ring.

Figure 18: the two channels corresponding to an edge as it leaves a node of V.

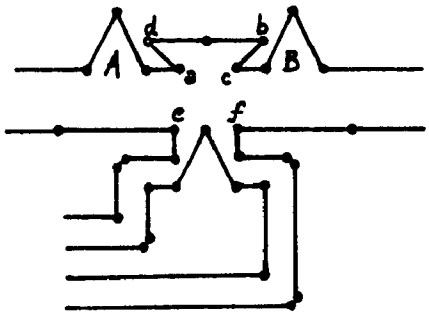
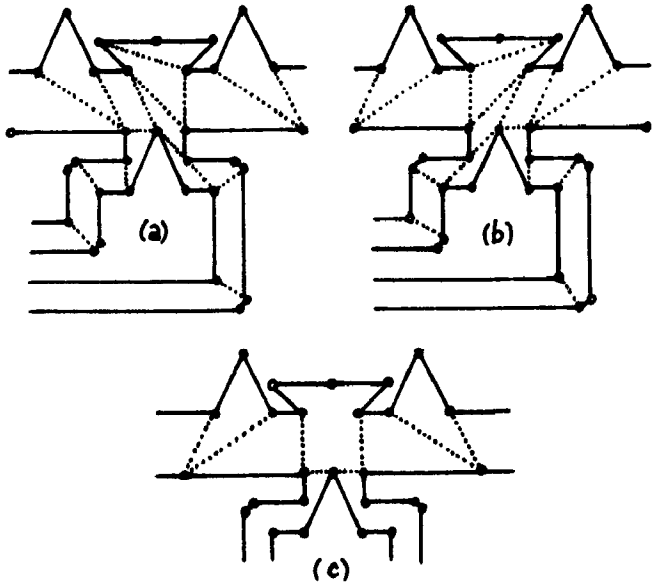
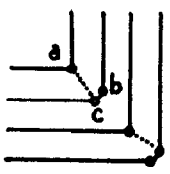


Figure 19.



The two channels corresponding to an edge will follow the edge in the planar embedding of G in a step-like fashion. At each corner of a channel exactly one of the two chords must be used (ab or ac in Figure 20). Because the start of the channels must be as in Figure 19(a) or (b) the two channels corresponding to an edge will have opposite choice of chord at every corner (see example in Figure 20). Parity can be changed by adding an extra node to each of the channels as in Figure 21.

Figure 20.



Replace each node of C in G by the structure shown in Figure 22.

Figure 21: changing parity.

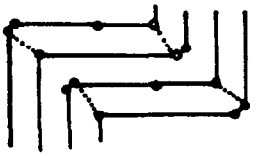
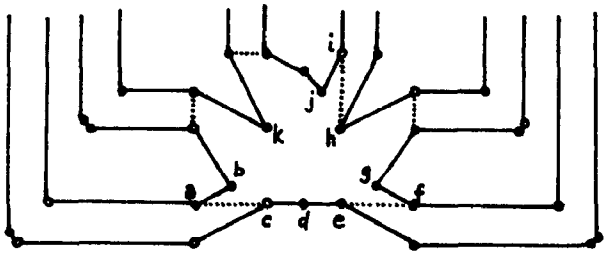


Figure 22: clause structure.



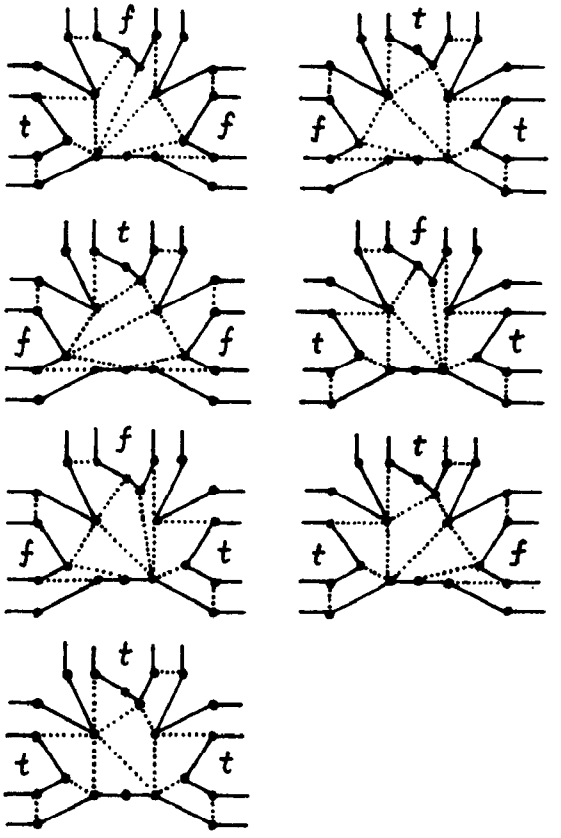
Figures 22 and 23 show the possible decompositions of the channels entering the cluse structure, the correspondence of these decompositions to true/false settings of the variables, and the decompositions (when possible) of the clause structure. Note that the convention as to which decomposition of a pair of channels corresponds to true and which to false depends on the position of the channels relative to the clause structure they join--so that parity changes, as shown in Figure 21, will in general be needed. Figure 22 shows the situation when none of the variables are true. In this case chord ij can be in a convex quadrilateral only with vertices c,d,e or h. Forming quadrilateral $jihd$ leaves two odd-sized polygons. Thus chords jc and hc, or je and he, or jc and he must be used. But then it is not possible to make convex quadrilaterals containing ab and gf. Thus the clause structure cannot be decomposed into convex quadrilaterals.

This completes the description of the polygonal region which can be decomposed into convex quadrilaterals if and only if the original formula was satisfiable. The construction can be carried out in polynomial time ■

Acknowledgements.

I thank Jack Edmonds for many stimulating ideas. I thank the members of the theory group at the University of Toronto, and especially Steve Cook, for helpful comments. This work was begun for Alain Fournier's computational geometry seminar.

Figure 23.



References.

[AHU] A.V. Aho, J.E. Hopcroft, and J.P. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading MA, 1974.

[CD] B. Chazelle and D. Dobkin, Decomposing a polygon into its convex parts, Proc. 11th Annual ACM Symp. on Theory of Computing, 1979, 34-48.

[EOW] H. Edelsbrunner, J. O'Rourke, E. Welzl, Stationing guards in rectilinear art galleries, Computer Vision, Graphics, and Image Processing 27, 1984, 167-176.

[F] S. Fisk, A short proof of Chvátal's watchman theorem, J. Combinatorial Theory B 24, 1978, 374.

[GJ] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, San Francisco, 1979.

[GJPT] M.R. Garey, D.S. Johnson, F.P. Preparata and R.E. Tarjan, Triangulating a simple polygon, Information Processing Letters 7, 1978, 175-179.

[K] J.M. Keil, *Decomposing Polygons into Simpler Components*, Ph.D. thesis, Dept. Computer Science, U. of Toronto, 1983.

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

[L] D. Lichtenstein, Planar satisfiability and its uses, SIAM J. Computing 11, 1982, 329-343.

[Ln] A. Lingas, The power of non-rectilinear holes, Proc. 9th Coll. on Automata, Languages, and Programming, Lecture Notes in Computer Science 140, Springer-Verlag, New York, 1982, 369-383.

[LPRS] A. Lingas, R.Y. Pinter, R.L. Rivest, A. Shamir, Minimum edge length decomposition of rectilinear polygons, unpublished manuscript.

[O] J. O'Rourke, An alternate proof of the rectilinear art gallery theorem, J. Geometry 21, 1983, 118-130.

[PS] W. Paul and J. Simon, Decision trees and random access machines, Logic and Algorithmic, Monograph 30, L'Enseignement Mathematique, 1980.

[PLLML] L. Pagli, E. Lodi, F. Luccio, C. Mugnai, W. Lipski, On two dimensional data organization 2, Fundamenta Informaticae, Vol. 2, No. 3, 1979.

[S] J.-R. Sack, An $O(n \log n)$ algorithm for decomposing simple rectilinear polygons into convex quadrilaterals, Proc. 20th Allerton Conf., 1982, 64-74.

[SH] M.I. Shamos and D.J. Hoey, Geometric intersection problems, Proc. 17th Annual IEEE Symp. on Foundations of Computer Science, 1976, 208-215.

[T] G.T. Toussaint, Pattern recognition and geometric complexity, 5th Internat. Conf. on Pattern Recognition, 1980, 1324-1347.

[Ta] R.E. Tarjan, *Data Structures and Network Algorithms*, CBMS-NSF Regional Conf. Series in Applied Math., SIAM, Philadelphia, 1983.