Folding a Simple Polygon:
A Paradigm for Computational Geometry

Ali A. Kooshesh
Department of Computer Science
Western Kentucky University
Bowling Green, KY 42101
kooshesh@cs.wku.edu

Bernard M. E. Moret
Department of Computer Science
University of New Mexico
Albuquerque, NM 87131
moret@cs.unm.edu

We describe a linear-time algorithm that folds a simple polygon decomposed into elementary convex regions (e.g., triangles or quadrilaterals) into one of its regions. Using this technique, we develop simple linear-time algorithms, based on the degree sequence of the boundary vertices of the polygon, to color the convex decomposition of the polygon and to construct its dual tree.

1 Introduction

We describe a linear-time algorithm that folds a simple polygon decomposed into convex k-gons into a single k-gon; for instance, we can fold a triangulated simple polygon into a single triangle. Using this technique, we can improve and simplify a number of algorithms that deal with convex decompositions of simple polygons into k-gons, as well as offer a structural characterization of various attributes of such polygons. Such characterizations include a very simple proof of Chvàtal's Watchman Theorem [3] and of its rectilinear version due to Kahn *et al.* [5]. We describe optimal algorithms, based on the degree sequence of the boundary vertices of the given simple polygon and its convex decomposition, to construct the dual tree of the decomposition in linear-time. The same algorithm, with minor modification, can be used to reconstruct the decomposition from its degree sequence; Beyer *et al.* [1] gave a similar algorithm for maximal outerplanar graphs, but our approach is simpler and more general. Parts of this work, limited to triangulated simple polygons, appeared in an earlier conference proceedings [6].

2 Convex Decompositions of Simple Polygons

A large number of algorithms in computational geometry rely on the convexity of the regions they manipulate. For this and other reasons, decomposing a simple, but non-convex polygon into convex regions has a been a problem of great interest in the field. The most useful convex decomposition of a simple polygon is through the addition of diagonals; since some of the convex pieces may have to be triangles, a regular partitioning must be a triangulation. Numerous algorithms have been devised for triangulating a simple polygon, culminating with Chazelle's linear-time algorithm [2].

When the polygon is rectilinear, i.e., when all edges of the polygon are parallel to one or the other of the axes, it admits a regular partition (through the addition of diagonals) into convex quadrilaterals, as was shown by Kahn *et al.* [5]. Certain simple polygons also admit regular partitions into k-gons, although no simple characterization of these polygons exists to our knowledge. We are interested in manipulating these regular convex partitions through a basic operation that we call *folding*.

3 Folding a Simple Polygon

The *dual tree* of the regular convex partition of a simple polygon is the tree formed by associating a vertex with each convex k-gon and connecting two vertices exactly when their associated convex k-gons share an edge. That the resulting graph is a tree is easily seen: it is obviously connected and, if it had any cycle, this would imply that the original decomposition of the polygon had a vertex not located on the perimeter and thus that it made use of additional segments rather than just diagonals. A *fan* is a sequence of adjacent convex k-gons with a common vertex, the *center* of the fan; the *ribs* of a fan are the diagonals emerging from its center; finally, the *base-path* of a fan is the simple path $v_0, v_1, \ldots, v_j$ defined by the traversal of vertices on the perimeter of the fan, starting and ending with vertices adjacent to the center of the fan. Note that, in an isolated fan, the degree of vertices v_1 through v_{j-1} must be either 2 or 3, with exactly $k-3$ vertices of degree 2 between any two vertices of degree 3; vertices of degree 3 are connected to the center of the fan by a diagonal, while vertices of degree 2 are not. A fan in which neither v_0 nor v_j has degree 3 is called *maximal*. A *sleeve* is a union of adjacent maximal fans such that the dual tree is a simple path. Figure 1 illustrates these concepts for triangulations.

We fold a sleeve by folding one of its end-fans, thereby reducing the number of fans by one, and repeating the process until a single k-gon results; we fold a polygon by folding subpolygons that form sleeves. Let an end-fan of a sleeve have center c and base-path $v_1, \ldots, v_j$; we fold this fan by folding it along its diagonals, all of which have c as one endpoint. The fan is thus folded into its last convex k-gon; if the just folded fan was not the last fan left in the sleeve, we now fold along diagonal $\{c, v_j\}$, thereby reducing the sleeve by one fan. We then repeat this process until the entire sleeve is folded into one convex k-gon. To fold a simple polygon, we effectively traverse its dual tree from its leaves (the dual tree need not be known, but it facilitates the description). When the traversal reaches a dual-tree vertex, y, of degree three or higher, we fold the sleeve, the dual tree of which is the simple path from the leaf to y; as a result, the degree of y decreases. Repeating this process eventually folds the polygon into a single convex k-gon. The process is illustrated for a triangulated simple polygon in Figure 2.

4 Coloring Results

Lemma 1. The folding algorithm partitions the vertex set of any simple polygon partitioned into convex k-gons into k equivalence classes (induced by the relation "is identified with") in such a way that, for any $2m+1$ consecutive vertices ($m < k$), $p_{i-m}, \ldots, p_{i-1}$, $p_i, p_{i+1}, \ldots, p_{i+m}$ (indices are modulo n), on the boundary of the polygon, p_{i-j} and p_{i+j}, $1 \leq j \leq m$, are in the same equivalence class if and only if the degree of p_i is odd and the degree of all p_{i-j} and p_{i+j}, $1 \leq j < m$, is even. □

Proof: That the relation "is identified with" on the vertex set of any simple polygon is an equivalence relation is immediate. At a vertex of even (respectively, odd) degree, the number of diagonals—and hence the number of foldings—is even (respectively, odd). With an odd number of foldings around vertex p_i, vertex p_{i-1} gets identified with vertex p_{i+1}, whereas an even number of foldings identifies vertex p_{i+1} with a vertex across the first diagonal emanating from p_i (if any). In general, with an odd number of foldings, vertex p_{i-j} gets identified with vertex p_{i+j}, but a vertex of odd degree interrupts the folding pattern by introducing its own mirror.

Q.E.D.

Let $\{C_1, C_2, \ldots, C_k\}$ be the partition induced by the relation "is identified with" on the vertex set of a simple polygon. Because we never fold an edge, but fold along edges, we have the immediate corollary.

Corollary 1. No two vertices of C_i, $1 \leq i \leq k$, are adjacent. In particular, the vertex set of any simple polygon partitioned into convex k-gons is two-colorable for k even, three-colorable for k odd. □

This corollary allows us to give a particularly simple proof of a theorem of Chvàtal [3]. Our proof is similar to that of Fisk [4], in that we both use the 3-colorability of a triangulated simple polygon. Our folding technique not only gives us a very clear proof of 3-colorability, but also shows how to reduce the problem of watching over the interior of a triangulated simple polygon to that of watching over the interior of a single triangle.

Corollary 2. [Chvàtal's Watchman Theorem] Let $P = (V, E)$ be a simple polygon with $|V| = n$. There exists a subset $V' \subset V$, with $|V'| \leq \lfloor \frac{n}{3} \rfloor$, such that any point internal to P is visible from at least one vertex in V'. □

Proof: Fold the triangulated polygon into one triangle and choose the vertex that represents the equivalence class with fewest members. This class contains at most $\lfloor \frac{n}{3} \rfloor$ vertices. Any point interior to the polygon is in a triangle, and every triangle has one vertex in the chosen class; therefore, any interior point is visible from at least one vertex.

Q.E.D.

With the same technique, we can prove that $n/4$ vertices suffice to watch over the interior of a rectilinear simple polygon.

Corollary 3. [Kahn *et al.* Watchman Theorem for Rectilinear Polygons] Let $P = (V, E)$ be a rectilinear simple polygon with $|V| = n$. There exists a subset $V' \subset V$, with $|V'| \leq \lfloor \frac{n}{4} \rfloor$, such that any point internal to P is visible from at least one vertex in V'. □

Proof: From the results of Kahn *et al.*, we know that P can be partitioned into convex quadrilaterals. Fold this partition into one quadrilateral and choose the vertex in that quadrilateral that represents the equivalence class with fewest members. This class contains at most $\lfloor \frac{n}{4} \rfloor$ vertices. Any point interior to the polygon is in a quadrilateral, and every quadrilateral has one vertex in the chosen class; therefore, since the quadrilaterals are convex, any interior point is visible from at least one vertex.

Q.E.D.

As we fold a k-gon along a diagonal, we can only fold it in one way; indeed, no matter where we start the folding process, the resulting partition is unique.

Corollary 4. The partition of the vertices of the simple polygon induced by the folding procedure is unique. □

The easy proof is omitted.

In a previous paper [7], we showed that producing a three-coloring for a triangulated simple polygon is trivial if it is implicitly based on folding. The algorithm requires as input the parity of each vertex and produces a 3-coloring in linear time. We can extend this algorithm to one that colors with k colors the vertices of a simple polygon partitioned into convex k-gons and that also runs in linear time with no other information than the parity of each vertex.

To simplify the matter, we define a class that provides two operations, `next()` and `reverse()`. The class has an internal state, called `direction`, that can either be clockwise or counter-clockwise. Initially, the value of this variable is clockwise. Function `reverse()`, when called, changes the state of this variable from clockwise to counter-clockwise and vice versa. Function `next(`k`)`, when called, returns a value in the range $[1, \ldots, k]$, which is the successor of the value it returned last time it was called. The first time it is called, it returns 1. When in clockwise (respectively, counter clockwise) state, the successor of i is $i+1$ (respectively, $i-1$) unless i is equal to k (respectively, 1) in which case its successor is 1 (respectively, k). The class is easily implemented as a circular doubly-linked list of k nodes with internal variables for the direction and the current pointer.

Let P be a simple polygon partitioned into convex k-gons and let its perimeter be $p_0, p_1, \ldots, p_n$. The following linear-time algorithm colors the vertex set of P using k colors.

for $i = 0$ to n do
begin
  if odd($\deg(p_i)$)
    then `reverse()`
  Color(p_{i+1}) $\leftarrow$ `next(`k`)`
end

We use induction to prove the correctness of the algorithm. The algorithm correctly k-colors the vertices of a k-gon. Let P be some simple polygon partitioned into convex k-gons and let $k-2$ vertices, $p^1, \ldots, p^{k-2}$, external to P, and edges $\{p_i, p^1\}$, $\{p^{k-2}, p_{i+1}\}$, and $\{p^l, p^{l_1}\}$ for $l = 1, k-2$, be added, where p_i and p_{i+1} are two consecutive vertices of P. Without loss of generality, assume that p_{i-1} and p_i of P have been colored 1 and 2 respectively. Tables 1 and 2, where $C(p)$ denotes the color of vertex p, demonstrate that a legal coloration of P will result in a legal coloration of $P \cup \{p, p_i\} \cup \{p, p_{i+1}\}$ (the indices are modulo $n+1$).

Table 1: Before adding $p^1, \ldots, p^{k-2}$

p_i	p_{i+1}	$C(p_{i+1})$	$C(p_{i+2})$
odd	odd	1	2
odd	even	1	k
even	odd	3	2
even	even	3	k’

Table 2: After adding $p^1, \ldots, p^{k-2}$

p_i	p_{i+1}	$C(p^1), \ldots, C(p^{k-2})$	$C(p_{i+1})$	$C(p_{i+2})$
even	even	3,$\ldots, k$	1	2
even	odd	$3, \ldots, k$	1	k
odd	even	$1, k, k-1, \ldots, 4$	3	2
odd	odd	$1, k, k-1, \ldots, 4$	3	k'

In the tables, we have used k' to denote $(4 \bmod k) + 1$. The value of k' is 4 if $k > 3$ and 1 if $k = 3$. When $k = 3$, $p^1, \ldots, p^{k-2}$ reduces to p^1 $1, k, k-1, \ldots, 4$ becomes 1. Also, notice that after the addition of the vertices, $p^1, \ldots, p^{k-2}$, the value of the state variable `direction` at p_{i+2} remains the same as it was before the addition of these vertices.

That the algorithm runs in linear-time is obvious from its description and the definition of functions `reverse()` and `next(`k`)`.

5 Branching in the Dual Tree

A k-gon is *branching* if three of its edges are diagonals of the convex decomposition. The branching k-gons in a give rise to vertices of degree three or higher in the dual tree; in other words, a convexly partitioned simple polygon is composed of sleeves attached to branching k-gons. Since constructing the dual tree of a sleeve is very simple, the identification of branching k-gons (or, equivalently, the identification of branching nodes when reconstructing the dual tree) is the crucial part of an algorithm that reconstructs the dual tree from its degree sequence. Note that branching k-gons are characteristic of convex decompositions, not directly of the polygons themselves; for instance, while every convex polygon can be triangulated as a single fan, it can also be triangulated with branching triangles. Note also that many non-convex polygons can be triangulated without branching triangles; somewhat unexpectedly, perhaps, monotone polygons may require branching triangles.

Lemma 2. A simple polygon must contain a branching k-gon in any convex decomposition into k-gons only if there exists a triple of vertices of the polygon, (x, y, z), each visible from the other two, such that, for each choice of one these three vertices, there exists at least one vertex on the perimeter between the other two vertices that is not visible from the chosen vertex. □

We omit the easy proof. Note that the three vertices of the characterization can, but need not be, vertices of a branching k-gon: all that the theorem asserts is that the presence of these three vertices is mandated by the presence of a branching k-gon. We can refine our characterization.

Lemma 3. Every simple polygon can be triangulated in such a way that each branching triangle in the triangulation (if any) has at least two reflex vertices. □

A restricted version of this result was proved by Kooshesh *et al.* [8] under a slightly different guise: they proved that every simple polygon has a triangulation in which a triangle that has at least two vertices on the convex hull of the polygon must have two consecutive vertices on the hull—so that no branching triangle can have two vertices on the convex hull. Using Chazelle’s convex decomposition theorem [2], which adds diagonals between reflex vertices to decompose a simple polygon into convex pieces, we can easily turn this restricted result into a proof of our theorem.

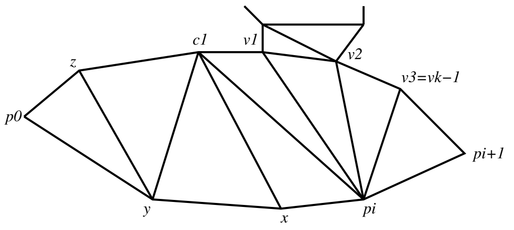
6 Reconstructing the Dual Tree

We present a linear-time algorithm, based on folding, that walks the perimeter of the polygon and, on the basis of the degree sequence, identifies all branching k-gons; in addition, our algorithm (with a very small modification) reconstructs the convex decomposition in linear time from the degree sequence. Our algorithm is based on balancing; it begins by identifying a sequence of $k-2$ vertices of degree 2—thereby identifying a k-gon whose dual is a leaf in the dual tree; it then walks the perimeter until it encounters another such sequence, recording on the way (in terms of degrees) what it sees of the (maximal) fans to which the vertices belong. On reaching the leaf in the dual tree, it starts matching the information it has accumulated with what it encounters on the other side of the perimeter. If no branching k-gon exists, the polygon is a simple sleeve and the degrees predicted from the information gathered on one side must exactly match the actual degrees of the vertices on the other side. When a mismatch is found (a vertex with actual degree larger than its predicted degree), it indicates the presence of a branching k-gon; by keeping a stack of the vertices on the first side and popping the stack as matches are made from the second side, we can identify the vertices on the first side that belongs to the branching k-gon and thus construct the dual tree of the polygon from its degree sequence.

We present a detailed description of the reconstruction algorithm as applied to a triangulated simple polygon; in this special case, a single vertex of degree 2 denotes the end of a sleeve. In the process of constructing the dual tree, our algorithm uses a vertex-stack and a degree-stack. (The degree-stack is used to simplify the exposition; in practice, instead of pushing vertices on the degree stack, we link them together to form the nodes of the dual tree.) Let s^d_k denote the kth entry from the top of the degree-stack and s^v_k the kth entry from the top of the vertex-stack; our algorithm proceeds as follows. Let p_0 be a vertex of degree 2; we initialize the degree-stack by pushing a 2 on it and call the procedure illustrated below to work on vertex p_0. (Refer to the accompanying illustration, Figure 3.)

`construct-dual-tree(`p_{i-1}`)`

($d_i = 3$) In this case, p_i (represented by vertex x in the figure) is on the base-path of some fan with center c_1 and s^d_1 is the degree of c_1 in the sleeve p_0–p_i. Since p_i is diagonal-adjacent to c_1, we increment s^d_1 by one and push a dual-tree node on the vertex-stack to account for the triangle xc_1x', where x' is the polygon vertex that is adjacent to x (represented by vertex p_i in the figure.)



	Degree Stack	Vertex Stack
before processing x	$[3,3]$	$[y,y]$
after processing x	$[3,4]$	$[y,y,x]$
after processing p_i	$[3,5,3,3,3]$	$[y,y,x,p_i,p_i,p_i,p_i]$
after processing p_{i+1}	$[3,5,4,3]$	$[y,y,x,p_i,p_i,v_2,v_2]$

Figure 4: Constructing the dual tree of a TSP

($d_i > 3$) In this case, p_i (p_i itself in the figure) is the center of a fan. Thus p_i is also an end-vertex of the base-path of some other fan, say centered at c_1, such that c_1 lies on the left-chain and is diagonal-adjacent to p_i. To reflect this adjacency, we increment s_1^d by one. Then we push $d_i - 3$ threes on the degree-stack to account for vertices $v_{k-1}, \ldots, v_1$ (the vertices on the base path of p_i, excluding c_1), that are expected to be of degree three. Finally, we push $d_i - 2$ dual-tree nodes on the vertex-stack to represent the faces induced by the triangles incident to the center, p_i, and to be later identified with the vertices of its base-path.

($d_i = 2$) In this case, p_i (represented by p_{i+1} in the figure) is the last vertex on the base path of some fan. Let k be the smallest natural number such that $d_{i+k} > s_k^d$; if such a vertex does not exist, let $k = n - i$. For $j = 1, \ldots, k$, we discard entries $s_k^d, \ldots, s_1^d$ and $s_m^v, \ldots, s_1^v$ where $m = \sum_{t=1}^{j}(s_t^d - 2)$. If the degree-stack is empty, stop, s_1^v is the root of the dual-tree. Otherwise, we increment s_1^d to account for $\{p_{i+k}, q\}$, where q is the vertex whose degree is represented by s_1^d and then push $d_{i+k} - s_k^d - 1$ threes on the degree-stack, push $d_{i+k} - s_k^d$ copies of p_{i+k} on the vertex-stack, and call `construct-dual-tree`(p_{i+k}). Note that the triangle $p_{i+k} s_1^v q$, where q is the vertex whose estimated degree is on the top of the degree-stack before the adjustments, is a branching triangle. Thus the algorithm identifies a branching triangle as it visits its second vertex.

7 Application to Outerplanar Graphs

In a previous paper [6], we showed that our coloring algorithm extended to k-trees (see [9] for definitions of k-trees) and that our reconstruction algorithm extended to 2-trees and to maximal outerplanar graphs (where every bounded face is a triangle), matching a result of Beyer *et al.* [1] while applying to a larger class of graphs. In fact, the folding approach allows us to reconstruct in linear time any outerplanar graph (not just maximal ones) that has k-gons for all of its bounded faces. In particular, we conclude that isomorphism of such graphs is decidable in linear time.

8 Conclusion

We have introduced the idea of folding simple polygons, which has led us to some simple characterizations of such polygons and then to particularly simple formulations of algorithms to color or reconstruct such polygons, and to compute the dual tree of the regular convex decomposition, from their degree sequence. Moreover, our approach extends to 2-trees and regular outerplanar graphs (where every bounded face has the same number of edges). Thus the folding paradigm is seen as a useful approach to several problems in computational geometry and algorithmic graph theory.

References

[1] T. Beyer, W. Jones, and S. Mitchell, "Linear algorithms for isomorphism of maximal outerplanar graphs," *J. ACM* **26** (1979), 603–610.

[2] B. Chazelle, "Triangulating a simple polygon in linear time," *Disc. Comp. Geom.* **6** (1991), 485–524.

[3] V. Chvàtal, "A combinatorial theorem in plane geometry," *J. Combin. Theory Series B* **18** (1975), 39–41.

[4] J. Fisk, "A short proof of Chvàtal's watchman theorem," *J. Combin. Theory Series A* **24** (1978), 374.

[5] J. Kahn, M. Klawe, and D. Kleitman, "Traditional galleries require fewer watchmen," *SIAM J. Algebraic Discrete Methods* **4** (1980), 194–206.

[6] A.A. Kooshesh and B.M.E. Moret, "Folding a triangulated simple polygon: structural and algorithmic results," *Lecture Notes in Computer Science* **497** (1991), 102–110.

[7] A. A. Kooshesh and B.M.E. Moret, "Three-coloring the vertices of a triangulated simple polygon", *Pattern Recognition* **25** (1992), 443.

[8] A.A. Kooshesh, B.M.E. Moret, and L.L. Székely, "Improved bounds for the prison yard problem," *Congressus Numerantium* **76** (1990), 145–149.

[9] D.J. Rose, "On simple characterizations of k-trees," *Discrete Mathematics* **7** (1974), 317–322.