A better approximation algorithm for covering polygons with squares *

[Extended Abstract]

Grażyna Zwoźniak
Institute of Computer Science
Wrocław University, Poland
grazyna@ii.uni.wroc.pl

ABSTRACT

We study the problem of covering a polygon without any acute interior angles, using a preferably minimum number of squares. The squares are allowed to overlap and they must lie entirely within the polygon. We show an $O(n \log n + \mu(P))$ time algorithm which covers any n-vertex input polygon with at most $10.5n + \mu(P)$ squares, where $\mu(P)$ denotes the minimum number of squares required to cover P. In the hole-free case our algorithm runs in linear time.

1. INTRODUCTION

One of the fundamental topics of computational geometry is how to decompose polygons into simpler objects, such as triangles, squares, rectangles, convex polygons. Considered polygons can contain polygonal holes. If the objects are allowed to overlap, then we call the decomposition a *covering.*

Considerable attention has received the problem of covering the polygon P with a minimum number of squares, all internal to the polygon, whose union is P. One application for this problem is storage images [13].

The *rectilinear case*, i.e. when the polygon and the squares have sides that are vertical or horizontal, has been treated in several papers [1, 2, 11, 13]. Authors use a boolean (zero-one) matrix, where one represents a point inside the polygon and zero – a point outside it. A complexity is measured in terms of the number of points in the matrix, denoted p. For most practical applications $p \gg n$. Aupperle, Conn, Keil and O'Rourke [1] show that the rectilinear case is NP-hard for polygons containing holes. For the hole-free case they present an $O(p^{1.5})$ algorithm. Bar-Yehuda and Ben-Hanoch [2] provide a linear time algorithm for such the case. Scott and Iyenger [13] present an algorithm to find, in $O(n \log n)$ time, a minimal cover for polygons with holes. However, their algorithm does not yield a globally minimum cover. Morita [11] presents a parallel algorithm, which finds a minimal (not minimum) square cover for such polygons. The sequential running time of this algorithm is $O(p)$.

The problem of covering polygons with a minimum number of rectangles has also been treated in several papers [3, 5, 7, 8, 10]. One application for it is the fabrication of masks for VLSI chips. Culberson and Reckhow [12] show that covering orthogonal polygons with the minimum number of ortogonal rectangles is NP-hard, even when the given polygon is hole-free. Levcopoulos [7] presents an algorithm which covers the polygon P with $O(n \log n + \mu^{'}(P))$ rectangles in $O(n \log n + \mu^{'}(P))$ time, where $\mu^{'}(P)$ is the minimum number of rectangles needed to cover P. In [8] a different heuristic is presented, guaranteeing an $O(\log n)$ approximation factor in polynomial time ($\Omega(n^6)$), provided that the vertices of the polygon have polynomially bounded integer coordinates.

In our paper [15] we consider the problem of covering polygons with a minimum number of squares. Our research was motivated by the paper of Levcopoulos and Gudmundsson [9]. They present two algorithms which cover an arbitrary polygon P, without any acute interior angles, with squares. Algorithm A_1 covers P using $14\mu(P)$ squares in time $O(n^2 + \mu(P))$, where $\mu(P)$ is the minimum number of squares needed to cover P. Algorithm A_2 uses $12n + \mu(P)$ squares and the running time is $O(n \log n + \mu(P))$. In our paper we present algorithm which covers polygon P with at most $10.5n + \mu(P)$ squares in time $O(n \log n + \mu(P))$.

2. THE VORONOI DIAGRAM OF P

Let S be a set of points and segments in the plane, such that:

- segments intersect only at endpoints,
- two endpoints of each segment belong to the set S.

Every point is assigned to the nearest element from S. The set of points equidistant from at least two elements is the *generalized Voronoi diagram* $V(S)$. The diagram is the union of lines, half-lines, segments and sections of parabo

*Partially supported by Komitet Badań Naukowych, grant 8 T11C 044 19.

Let P be an arbitrary polygon without any acute interior angles, with n vertices and w concave vertices. $V(P)$ is the part of generalized Voronoi diagram $V(S)$ lying within P, where S consists of all edges and vertices of P. $V(P)$ partitions P into $n + w$ *regions*, where each segment and each concave vertex induces a Voronoi region. Every point lying in a region induced by d, where d is either an edge or a concave vertex, lies at least as close to d as to any other point of the boundary of P. Each *Voronoi circle* $c_V(p, r)$ with the center in $p \in V(P)$ and the radius r equal to the shortest distance from p to the boundary of P, lies entirely within P. The diagram can be computed in $O(n \log n)$ time [4, 6] for an arbitrary polygon and in linear time for a hole-free polygon [14].

3. PARTITIONING INTO CELLS

Let f be a Voronoi region induced by $g \in S$. If g is an edge of P, then we partition f into cells by drawing segments, called *sides* of cells, which connect Voronoi vertices of f with their perpendicular projections on g. The part of g bounding the cell is called its *base*. If g is a concave vertex of P, then we partition region f by drawing segments connecting all its Voronoi vertices with g. A cell may be a triangle, a trapezoid, an area bounded by two segments and a part of paraboloid (the cell induced by a concave vertex), or an area bounded by three segments and a part of paraboloid (the cell induced by an edge of P). Note that every cell is bounded by a proper Voronoi edge.

Let $\mathcal{T}$ be the set of all trapezoidal cells in P. A cell $k \in \mathcal{T}$ is a *funnel*, if the angle between the proper Voronoi edge and the shorter side of the cell is smaller than 135^o, and the length of its base is greater than the length of the shorter side; otherwise it is an *ordinary trapezoid*. If a cell $k \notin \mathcal{T}$, then k is called a *non-trapezoid*. In this paper $\mathcal{F}$ denotes the set of all funnels in P and $\mathcal{O}$ denotes the set of all ordinary trapezoids in P.

Two cells sharing a proper Voronoi edge are called a *pair of cells*.

THEOREM 1. *The number of proper edges in V(P) is at most 3n, where n is the number of vertices in the polygon P.*

COROLLARY 1. *The number of pairs of cells is not greater than 3n.*

We classify each pair of cells k_1k_2 as follows:

1. a pair of trapezoids; k_1 and k_2 are induced by edges of P, Fig. 1(a);
2. a pair of non-trapezoids:
   (a) a pair of triangles; k_1 and k_2 are induced by edges of P, Fig. 1(b);
   (b) a pair of triangles; k_1 and k_2 are induced by concave vertices of P, Fig. 1(c);
   (c) a pentagon; k_1 is induced by a concave vertex of P and k_2 is induced by an edge of P, Fig. 1(d).

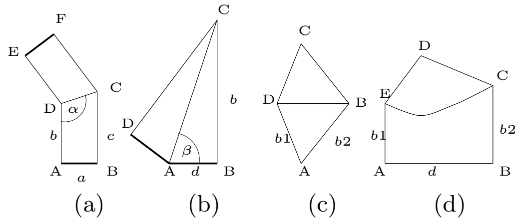

Figure 1: (a) A pair of trapezoids. (b) Triangles induced by edges of P. (c) Triangles induced by concave vertices of P. (d) A pentagon.

A *sequence of trapezoids* is a maximal sequence $p_1, \ldots, p_k$, $k \geq 2$ such that:

- $p_i \in \mathcal{T}$, $i = 1, \ldots, k$,
- the longer side of p_i is the shorter side of p_{i+1}, $i = 1, \ldots, k-1$,
- $p_i \in \mathcal{F} \Rightarrow p_{i+1} \in O$, $i = 1, \ldots, k-1$.

We say that a trapezoid t is *adjacent* to a pair of non-trapezoids n_1n_2 if a side of either n_1 or n_2 is the shorter side of t. Even if two sides of t have the same length, we call one of them the shorter side (we fix at the beginning which one is shorter). Note that at most four trapezoids may be adjacent to n_1n_2.

4. THE ALGORITHM

Our algorithm is based on the following scheme.

1. Compute $V(P)$.
2. Partition each Voronoi region into cells.
3. Assign each cell to one of the groups as follows:
   (a) Fix sequences of trapezoids and then divide each of them into groups consisting of two consecutive cells (if the number of trapezoids in the sequence is odd, then the last group has three cells).
   (b) Assign each non-trapezoid n_1 to the group induced by the pair of non-trapezoids n_1n_2 (n_2 shares with n_1 a proper Voronoi edge).
   (c) Assign the remaining trapezoids to one of the following group:
      i. if t is adjacent to a pair of non-trapezoids n_1n_2, then add t to the group induced by n_1n_2;
      ii. if t is a funnel that has not been assigned to any sequence of trapezoids and any pair of non-trapezoids, then t is a separate group.
4. Cover each group with squares.

The following lemma guarantees that each cell will be covered by our algorithm.

LEMMA 1. *Each cell belongs to exactly one of the groups.*

In the fourth step our algorithm covers groups of cells. At the beginning it covers all funnels, and then the remaining cells in the groups. The covering of funnels may require usage of large number of squares. Fortunately, we can make use of their appearance in the group and reduce the number of squares that cover the remaining cells. In [15] we give a detailed analysis of all the cases. We also prove that our algorithm uses $\mu(P)$ squares plus - perhaps - $O(n)$ additional squares. In average the surplus of squares is not greater than 3.5 per one pair of cells.

THEOREM 2. *Our algorithm covers* P *with at most* $10.5n+\mu(P)$ *squares.*

There are two main reasons why our algorithm uses less squares than the algorithm presented in [9]. Firstly, we can better cover some pairs of non-trapezoids. Secondly, if we consider groups of cells, we can use a smaller number of larger squares that lie within the polygon P.

4.1 Time analysis

We can construct $V(P)$ in $O(n \log n)$ time, if P is an arbitrary polygon [6, 4], and in $O(n)$ time, if P is a hole-free polygon [14]. Because there are at most $6n$ cells (see Corollary 1), we can split Voronoi regions into the cells in $O(n)$ time, and then assign each cell to the group in $O(n)$ time.

In full version of our paper [15] we show that each group of cells may be covered in $O(s)$ time, where s is the number of squares used for covering the considered group of cells. There are at most $6n$ cells, so P may be covered in $O(n+s) = O(n+\mu(P))$ time.

5. REFERENCES

[1] L. J. Aupperle, J. M. Keil, J. O'Rourke. Covering orthogonal polygons with squares. In *26th Annual Conference on Communication, Control and Computing*, pages 97–106, 1988.

[2] R. Bar-Yehuda, E. Ben-Hanoch. A linear-time algorithm for covering simple polygons with similar rectangles. In *International Journal of Computational Geometry and Applications, vol. 6, no. 1*, 1996.

[3] B. M. Chazelle. Computational geometry and convexity. *Ph. D. Thesis, Department of Computer Science, Yale University, New Haven, C.T., Carnegie-Mellon Univ. Report CS-80-150*, 1979.

[4] S. Fortune. A sweepline algorithm for Voronoi diagrams. In *2nd Annual ACM Symposium on Computational Geometry*, pages 313–322, 1986.

[5] A. Hegedus. Algorithms for covering polygons by rectangles. In *Computer Aided Design, vol. 14, no. 5*, 1982.

[6] D. G. Kirkpatrick. Efficient computation of continuous skeletons. In *20th Annual IEEE Symposium on Foundation of Computer Science*, pages 18–27, 1979.

[7] C. Levcopoulos. A fast heuristic for covering general polygons with rectangles. In *Fundamentals of Computer Theory, Cottbus, GDR, LNCS 199, Springer-Verlag*, 1985.

[8] C. Levcopoulos, J. Gudmundsson. Close approximation of minimum rectangular coverings polygons. In *FST-TCS'96, LNCS 1180, Springer-Verlag*, pages 135–146, 1996.

[9] C. Levcopoulos, J. Gudmundsson. Approximation algorithms for covering polygons with squares and similiar problems. In *RANDOM'97, LNCS 1269, Springer-Verlag*, pages 27–41, 1997.

[10] C. Levcopoulos, A. Lingas Covering polygons with minimum number of rectangles. In *Symposium of Theoretical Aspects of Computer Science, LNCS 166, Springer-Verlag*, pages 63–72, 1984.

[11] D. Morita. Finding a minimal cover for binary images: an optimal parallel algorithm. *Technical Report No. 88-946, Department of Computer Science, Cornell University*, 1988.

[12] R. A. Reckhow, J. Culberson. Covering a simple orthogonal polygon with a minimum number of orthogonally convex polygons. In *3rd Annual ACM Symposium on Computational Geometry*, pages 268–277, 1987.

[13] D. S. Scott, S. S. Iyenger. Tid: a translation invariant data structure for storing images. In *Communications of the ACM, vol. 29, no. 5*, 1986.

[14] J. Snoeink, C. A. Wang, F. Chin. Finding the medial axis of a simple polygon in linear-time. In *International Symposium on Algorithms and Computation, LNCS 1004, Springer-Verlag*, pages 382–391, 1995.

[15] G. Zwoźniak. A better approximation algorithm for covering polygons with squares. *Technical Report, Institute of Computer Science, University of Wrocław*, 1998.