An O(Nlog*N) Time Algorithm for Covering Simple Polygons with Squares

(Extended Abstract)

Reuven Bar-Yehuda* *and* Eyal Ben-Chanoch

Computer Science Department
Technion - IIT, Haifa 32000, Israel

Abstract

We study the problem of covering a simple polygon with a minimum number of squares, possibly overlapping, all internal to the polygon. The problem has applications in VLSI mask generation, incremental update of raster displays, and image compression. A polygon is specified by the (cyclically ordered) sequence of its *n* vertices. We give an algorithm for covering a simple polygon with squares in $O(n\log^* n)$ time. In the special case where the coordinates of the vertices are integers, a simplified version of our algorithm, has a time complexity which is linear in the size of a bit-map representation. We also consider non-simple polygons. In this case the problem is known to be NP-Hard. We show that a factor 2 approximation can be found in polynomial time.

1. Introduction

We consider the problem of finding a minimum square cover of an orthogonal polygon. An *orthogonal polygon* is a polygon whose edges are either horizontal or vertical. A *square cover* of a polygon *P* is a collection of squares contained within *P*, whose union exactly covers *P*. A *minimum square cover* is one with the minimum number of squares. Obviously, we can concentrate only on maximal squares. A square in the polygon is said to be *maximal*, if it is not contained by any larger square in the polygon.

Our research was motivated by the paper of Scott and Iyenger [SI86] where they argued that square covering is an efficient method to store images, mainly in comparison to the known quad-trees approach [SL87], [HS79]. Moreover, they revealed some of the characteristics of this method such as simplicity, limited space demands and invariance in the plane under shifts and rotations. In their work they developed a practical heuristic algorithm to find a square cover (not necessarily minimal), empirically illustrating better performance than all of the quad-tree

* The research was supported in part by Technion V.P.R Fund - Albert Einstein Research Found.

methods they tested.

Aupperle at al. [ACKO88] noted another practical application which we quote. The *medial axis* (also called the *symmetric axis* or *skeleton*) of a polygon is the locus of centers of maximal disks contained in the polygon. When specialized to the L_∞ metric for applications to digital images, the medial axis is the locus of centers of maximal squares of odd side length [RK82]. The digital medial axis transform (*MAT*) is used for picture compression: simple images may be covered by few squares, and easily reconstructed from the *MAT* [WBR86].

Additional applications are mentioned by Morita [M89], such as remote sensing, VLSI mask generation, incremental update of raster displays, and object representation for sequential frames of dynamic polygonal scene.

1.2 Related Work

The previous work on the square cover problem used a bit-map representation for the input polygon [AO81], [SI86], [ACKO88], [M89]. A *bit−map* representation of a polygon, or sometimes referred to as a *digitized image*, is a zero-one matrix, where one represents a point within the polygon and zero otherwise. In respect to this representation, we measure the complexity in terms of the number of points in matrix p.

Recently Aupperle, Conn, Keil and O'Rourke [ACKO88] showed that the minimum square cover problem is NP-Hard for images containing holes. In the case where the image is hole-free, they provided an $O(p^{1.5})$ time algorithm. They translate the problem to covering chordal graphs by cliques (equivalent to find minimum vertex cover). Each node within the graph is associated to a pixel in the bit-map. Two nodes are connected by an edges iff the corresponding pixels are within the same some maximal square. Clearly, the nodes of a maximal clique correspond to the pixels of a maximal square. Since the image is hole-free, all cycles in the graph are triangulated, therefore, the algorithm of Gavril [G72] for covering chordal graphs by minimum number of cliques can be used. To decrease the size of the graph they generate a reduced graph, where each maximal clique (square) is translated into a single node.

A square cover is called *minimal*, if has no smaller subset that forms a cover. Morita [M89] recently developed a parallel algorithm which finds a minimal square cover for bit-maps which may contain holes. The sequential time of this algorithm is $O(p)$.

1.3 Bit-maps vs Segment Representation

It seems that the original motivation to study the square cover problem was derived from the area of image processing and other related fields. This may explain why all the previous work concern only polygons with integer coordinates. Furthermore, the time complexity was measured in terms of the input bit-map size p.

For many problems concerning polygonal shapes, the input is chosen to be a segment representation, i.e. the polygon is specified by a cyclically ordered sequence of its n vertices. We argue that this representation should be preferred, not only theoretically, but also for all of the mentioned practical applications.

In order to illustrate that a bit-map based algorithm may, in the worst case, be exponential (or at the most pseudo-polynomial) in the segment representation input size, consider the following example: The input polygon ((0,0), (0,y), (1,y), (1,0)) which consumes only $O(logy)$ space, requires at least $O(y)$ space for the bit-map representation. On the other hand, it may be worthwhile to spend an $O(p)$ time in translating a bit-map representation into a segment representation, since $p > \Omega(n)$, and for most practical applications $p \gg n$. This translation can be implemented on-line, in reading image files. This method requires only $O(n \cdot \log p)$ bits rather than $O(p)$.

The usage of the chordal graph method is elegant and natural for bit-maps. In the segment representation the associated chordal graph, and even the reduced one, may have infinite cardinality. In our method, the idea of using local optimization approach is kept i.e. selecting, iteratively, the next "essential" square. Local optimization algorithms differ in the policy adopted for the next local optimization step. The core of our algorithm is the chois of a specific policy derived by additional topological properties.

1.4 Outline of The Algorithm

In this subsection we give an intuitive description of our $O(n\log^* n)$ algorithm.

Imagine that in the beginning of the process you have a piece of black paper cut in the shape of the given input polygon. As the algorithm proceeds, we paint the covered areas with a grey chalk.

Let s be a maximal square in the polygon. Denote by $Black(s)$ the set of the black (uncovered) points in s. A square s is *maximal black* if there is no square s_1 such that $Black(s) \subset Black(s_1)$. A maximal black square s is *essential* if there exist a point such that s is the only square to contain it.

Algorithm A:
  While there exist essential square s
  Paint s grey.

Assume the paper is now all painted grey; then the set of the selected squares yields a minimum square cover. This fact is derived from the following local optimization invariant:

there exist a minimum square cover for the polygon which contain all of the previously selected squares.

A grey point is *reducible* if it does not included in any maximal black square. Now, for algorithm B, we have to get the scissors ready.

Algorithm B:
While there exist essential square s
paint s grey and cut all reducible regions.

Cutting reducible regions is harmless, and therefore the local optimization invariant remains valid.

The approach: select "essential" and eliminate "reducible", is the same for all special cases of the set-covering problem, including covering chordal graphs by cliques [G72] and vertex cover [NT75], [BE85]. Unlike those problems, in our problem, both, the number of squares (sets) and the number of points (set elements) are infinite. This fact would not affect the correctness of the local optimization approach, as long as an essential square can be found, and in each iteration the problem size (input+output) is reduced. In order to achieve this goal we concentrate on special type of essential squares called continuators. A *continuator* is a maximal square having a continuous intersection with the polygons' contour (for a wider family of squares containing a *knobs* see [AO81],[ACKO88]). The main idea is that if the residual paper does not contain reducible regions then any continuator is essential.

Algorithm C:
While there exist continuator s
if s is essential then paint s grey.
cut s reducible regions.

We prove that any simple polygon contains a continuator. This, together with the local optimization invariant, implies partial correctness for Algorithm C. By charging each operation to either input (vertices) or output (cover squares) we show that the number of iterations is $O(n+k)$. This algorithm can be implemented so after a preprocessing phase consisting mainly of triangulating the polygon, each output operation costs $O(1)$ time, while each input operation costs $(logn)$ time. This leads us to $O(n\log n+k)$ time complexity. Now, we restrict ourselves to a continuator with an additional topological property. This leads to an implementation with time complexity of $O(1)$ for output, but, $\ddot{o}(log^*n)$ for each input operation. Since the best known algorithm for triangulation (preprocessing) has $O(n\log^*n)$ time complexity [C90], the total time is reduced to $O(n\log^*n+k)$.

As it presented, the algorithm is output sensitive. Note that the output size may be even exponential in the input size (the polygon is a long rectangle). O'Rourke suggested a way to overcome this problem by changing the output representation. A set of output squares that yield a partitioning of a rectangle, would be replaced by this rectangle. Now, the output size is $O(n)$. Fortunately, our algorithm can be implemented, so each such a rectangle requires $O(1)$ operations.

Acknowledgment

We would like to thank Prof. Alon Itai for introducing us to the problem.

References

[A88] Aupperle, L. J., "*An Algorithm for Covering Polygons with Squares*," Draft, Dept. of Comp. Sci., Princeton Univ., Oct. 1988.

[ACKO88] Aupperle, L. J., H. E. Conn, J. M. Keil and J. O'Rourke, "*Covering Orthogonal Polygons with Squares*," 26^{th} Ann. Allerton Conf. Communication, Control and Computing, Urbana IL, 28-30 Sep. 1988.

[AO81] Albertson, O.M., and C.J. O'Keefe, "*Covering Regions with Squares*," SIAM J. Alg. Disc. Math., vol. 2, pp. 240-243, 1981.

[BE85] Bar-Yehuda, R., and S. Even, "*A Local Ratio Theorem for Approximating the Weighted Vertex Cover Problem*," Annals of Discrete Mathematics, vol. 25, pp. 27-46, 1985.

[C90] Chazelle, B., FIEfficient Polygon Triangulation, *Dept. of Coputer Scie. Princeton Univ., CS-TR-249-90, Feb. 1990.*

[G72] Gavril F., "*Algorithms for Minimum Coloring, Maximum Clique, Minimum Coverig by Cliques and Maximum Independent Set of Chordal Graphs*," Siam J. of Computing, vol 1 (2), pp 180-187, 1972.

[HS79] Hunter G. M., and K. Steiglitz, "*Operations on images using Quad-Trees*" IEEE Trans. Pattern Analysis Machine Intell. PAMI-1,2, 1979.

[M89] Morita, D., "*Finding a Minimal Cover for Binary Images: an Optimal Parallel Algorithm*," Information System Lab. 89CRD149, Aug. 1989. To appear in ALGORITHMICA.

[NT75] Nemhauser, G. L., and L. E. Trotter, Jr., "*Vertex Packing: Structural Properties and Algorithms*," Mathematical Programming, vol. 8, pp. 232-248, 1975.

[RK82] Rosenfeld. A., and A. C. Kak, "*Digital Picture Processing*," Academic Press, New York, 1982.

[SI86] Scott, D. S., and S. Iyenger, "*TID - Translation Invariant Data Structure of Storing Images*," Comm ACM vol. 29, pp. 418-428, 1986.

[SL87] Shu-Xiang L. and M. H. Loew, "*The Quad-Codes and Its Arithmetic*," Comm. ACM vol. 30, pp. 621-625, 1987.

[WBR86] Wu A. Y., S. K. Bhaskar, and A. Rosenfeld, "*Computation of Geometric Properties from the Medial Axis Transform in O(nlogn) Time*," Computer Vision, Graphics, and Image Processing, vol 34, pp. 76-92, 1986.