数理解析研究所講究録
第 522 巻 1984 年 53-61

Topics on Triangulation of Polygons

Tetsuo Asano*, and Takao Asano**
浅野哲夫　浅野孝夫
* Osaka Electro-Communication University,
Neyagawa, Osaka, 572 Japan,
** Faculty of Engineering, University of Tokyo,
Bunkyo-ku, Tokyo, 113 Japan

SUMMARY

In this paper we show that Ω(nlogn) operations are necessary to triangulate a polygonal region with n vertices which contains windows or holes. Also, we present a polynomial time algorithm for partitioning a polygonal region which may have a fixed number of windows into a minimum number of triangles.

1. Introduction

Concerning a problem of triangulating a polygonal region which may have windows or polygon holes, a number of papers have been presented. Main interests were (1) to devise an efficient algorithm for triangulating a polygonal region [1-4], (2) to analyze the complexity of the minimum number decomposition problem and the minimum edge length decomposition problem [5-7], and (3) to develop a polynomial time exact algorithm using a dynamic programming approach [8,9]. In this paper we show that $\Omega(n\log n)$ operations are necessary to triangulate a polygonal region with n vertices. Thus, the triangulation algorithm proposed by Garey, Johnson, Preparata, and Tarjan[1] which can be applied to not only simple polygons but also polygonal regions with windows is optimal within a constant factor. However, the lower bound for the polygon triangulation problem remained open. Also we present a polynomial time algorithm for partitioning a polygonal region which may have a fixed number of windows into a minimum number of triangles.

2. Lower Bound for Triangulation Problem

In this section we show that $\Omega(n\log n)$ operations are necessary to triangulate a polygonal region which may have windows. It is shown below that the sorting problem on n positive integers can be transformed in linear time into the problem of trangulating a polygonal region with 3n+4 vertices and n windows. It is well known that the lower bound for the sorting problem is $\Omega(n\log n)$.

Consider the set of n positive integers $x_1, x_2, \dots, x_n$. Let m be a minimum and M be a maximum among them. Here we can assume that any two of them are distinct. We construct a polygonal region P as follows. The external boundary of P is a triangle specified by three vertices (0, 0), (4M-2m-1, 2M-1), and (4M-2m-1, -2M+1). P contains n triangular windows, each consisting of three vertices $(2x_i, x_i)$, $(2x_i, -x_i)$, and $(2x_i+1, 0)$, $1 \leq i \leq n$. An example of such a polygonal region is illustrated in Fig. 1. This polygonal region possesses several triangulations. However, as is easily seen, the list of edges incident to the vertices $(2x_i+1, 0)$'s can be used to sort the numbers $x_1, x_2, \dots, x_n$ in O(n) time, for there are only two vertices visible from each vertex $(2x_i+1, 0)$ as shown in Fig. 2.

Here it should be noted that in the above proof we allow O(n) windows for the polygonal region. The lower bound is not known for the problem of triangulating a polygonal region which contains a fixed number of windows.

3. Minimum Number Triangulation of Polygonal Regions

The problem of partitioning a polygonal region which may contain windows into a minimum number of triangles is known to be NP-complete. However, for a polygonal region with a fixed number of windows, we can construct a polynomial time exact algorithm using a dynamic programming approach. As for the minimum edge length triangulation problem, G.T.Klincsek presented $O(n^3)$ time exact algorithm for polygons without any window using a dynamic programming approach. His method is based on the definition that a triangulation of a polygon is a maximal subset of chords in which no two of them cross each other. A similar but more precise algorithm is described in the book written by Aho, Hopcroft, and Ullman. However, it is easily seen that such a maximal subset of chords decompose a polygon with n vertices into n-2 triangles. Thus, we must develop a different algorithm for the minimum number triangulation problem. Fortunately, we have only to modify their method so that we allow triangles with the area being zero.

First, we present an $O(n^3)$ time algorithm for polygons, polygonal regions with no windows, where n is the number of vertices. Next, we show that there exists an $O(n^{3+2k})$ time exact algorithm for polygonal regions with k windows and n vertices.

Consider a polygon P with n vertices v_0, v_1, ... , v_{n-1}, in clockwise order. Let $P_{i,t}$ denote a subpolygon of P which is formed by P's edges (v_i, v_{i+1}), (v_{i+1}, v_{i+2}), ... , (v_{i+t-2}, v_{i+t-1}) and the segment (v_i, v_{i+t-1}), where we take all subscripts to be computed modulo n. A subpolygon $P_{i,t}$ is called a valid subpolygon when the segment (v_i, v_{i+t-1}) is an edge or chord of P, where a chord is defined to be a segment between two vertices of P which lies entirely within P. Notice that a chord may be on the boundary of P. By $p_{i,t}$ we denote the size of the minimum number triangulation of $P_{i,t}$ if $P_{i,t}$ is a valid subpolygon. Otherwise, $p_{i,t}$ is defined to be positive infinite.

The problem here is to compute $p_{0,n}$. In general, we can compute $p_{i,t}$ for a valid subpolygon $P_{i,t}$ based on the fact that in any triangulation of $P_{i,t}$ there exists a triangle that contains the side (v_i, v_{i+t-1}). Here we allow a dummy triangle the area of which is zero. For example, in the polygon $P_{0,5}$ shown in Fig. 3, the triangle (v_0, v_2, v_4) is a dummy triangle. Anyway, we have only to consider decompositions of $P_{i,t}$ by (v_i, v_k) and (v_k, v_{i+t-1}) which may be edges or chords of $P_{i,t}$, where $i+1 \leq k \leq i+t-2$. When $k = i+1$ or $k = i+t-2$, $P_{i,t}$ is decomposed into two parts, that is, in the former case into a triangle $(v_i, v_{i+1}, v_{i+t-1})$ and a subpolygon $P_{i+1,t-1}$, and in the latter case into a subpolygon $P_{i,t-1}$ and a triangle $(v_i, v_{i+t-2}, v_{i+t-1})$. When $i+2 \leq k \leq i+t-3$, $P_{i,t}$ is decomposed into three parts $P_{i,k-i+1}$, $P_{k,t-k+i}$, and a triangle (v_i, v_k, v_{i+t-1}), as shown in Fig. 4. In particular, if a triangle $(v_i, v_{i+1}, v_{i+t-1})$ is a dummy triabgle, then we have

$$p_{i,t} = p_{i+1,t-1}.$$

For the same reason, if a triangle $(v_i, v_{i+t-2}, v_{i+t-1})$ is a dummy triangle, we have

$$p_{i,t} = p_{i,t-1}.$$

Furthermore, a triangle (v_i, v_k, v_{i+t-1}), $i+2 \leq k \leq i+t-3$, may happen to be a dummy triangle. Therefore, we introduce another symbol $d_{i,j,k}$ defined by

$$d_{i,j,k} = 0 \quad \text{if } (v_i, v_j, v_k) \text{ is a dummy triangle,}$$

$= 1$ otherwise.

Using the symbol, we can compute $p_{i,t}$ by the following formula:

$$p_{i,t} = \min[\, p_{i+1,t-1} + d_{i,i+1,i+t-1},\ p_{i,t-1} + d_{i,i+t-2,i+t-1},\ \min_{i+2 \le k \le i+t-3} (p_{i,k-i+1} + p_{k,t-k+i} + d_{i,k,i+t-1})].$$

An efficient way to solve the triangulation problem follows from the above discussion. We make a table giving $p_{i,t}$, the size of the minimum number triangulation of $P_{i,t}$ for all i and t, $0 \le i \le n-1$ and $3 \le t \le n$. Since the solution to any given problem depends only on the solution to problems of smaller size, we can fill in the table in ascending order of t, that is, t = 3, 4, ... , n. In order to find a set of chords to triangulate P optimally, we have only to store an optimal chord to achieve the minimum number decomposition of each subpolygon $P_{i,t}$. Each $p_{i,t}$ can be computed in $O(n)$ time and the table size is $O(n^2)$. Thus, the complexity of the above described algorithm based on dynamic programming is $O(n^3)$.

Consider an example shown in Fig. 5 to explain the algorithm. Table 1 shows the costs $p_{i,t}$'s. Thus, we find that the polygon is decomposed into five triangles. Such a decomposition is derived as shown in Fig. 6.

Next, we propose an $O(n^{3+2k})$ time algorithm for partitioning a polygonal region with k windows into a minimum number of triangles. We already presented such an algorithm for the case of k = 0. We assume that there exists an $O(n^{3+2i})$ time exact algorithm for polygonal regions with i windows where $i < k$. Now, consider a polygonal region P with k windows. Our algorithm is based on the fact that in any triangulation there exists at least one triangle (v_i, v_j, v_k) such that (v_i, v_j) is an edge of a specified window of P and v_k is not any vertex of the window. When we decompose P by such a triangle we obtain a polygonal region P' with exactly k-1 windows. By the hypothesis we can find the minimum number triangulation of P' in

$O(n^{3+2(k-1)})$ time. The number of such triangles is less than n^2 even in the worst case, and we can obtain the representation of P' in O(n) time. We can enumerate such triangles in $O(n^3)$ time as follows: For each edge (v_i, v_j) of the specified window of P, find a set of vertices which are not on the window and visible from both v_i and v_j. Then, if v_k is such a vertex, (v_i, v_j, v_k) is a triangle required if both of (v_i, v_k) and (v_j, v_k) are chords.

Thus, we can find a minimum number triangulation of P by performing the process in $O(n^3)$ time and then solving at most n^2 subproblems each in $O(n^{3+2(k-1)})$ time. This leads to the algorithm required.

REFERENCES

[1] M.R. Garey, D.S. Johnson, F. Preparata, and R.E. Tarjan, "Triangulating a simple polygon", Information Processing Letters, Vol.7, June 1978, pp.175-180.

[2] A.A. Schoone and J. van Leeuwan, "Triangulating a star-shaped polygon", Tech. Rept. RUV-CS-80-3, University of Utrecht, April 1980.

[3] M. Shamos, "Geometric Complexity", PhD Thesis, Yale University, May 1978.

[4] E.Lloyd, "On Triangulations of a set of points in the plane", Proc. 18th FOCS, October 1977, pp.228-240.

[5] A. Lingas, "The Power of Non-Rectilinear Holes", in G. Goos and J. Hartmanis(eds.), Ninth International Colloquim on Automata, Languages and Programming, Aarhus, July 1982, Lecture Notes in Computer Science 140, Springer-Verlag, Berlin, 1982, pp.369-383.

[6] M.R.Garey and D.S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W.H.Freeman and Co., San Francisco, 1979.

[7] D.S. Johnson, "The Column of NP-Completeness: An Ongoing Guide", J. Algorithms, Vol.3, 1982, pp.182-195.

[8] G.T. Klincsek, "Minimal Triangulations of Polygonal Domains", Annals of Discrete Math. Vol.9, 1980, pp.121-123.

[9] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, "Data Structures and Algorithms", Addison-Wesley Publishing Co., 1983.

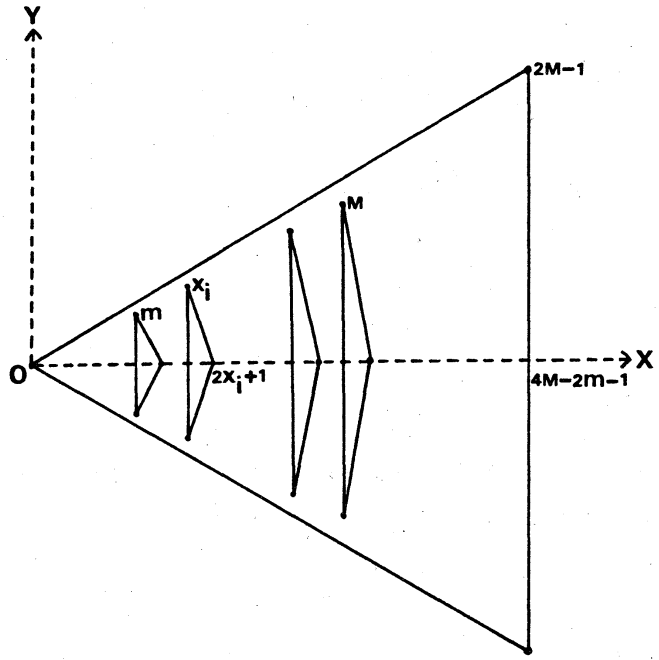

Fig. 1 A polygonal region P with 3n+3 vertices and n windows.

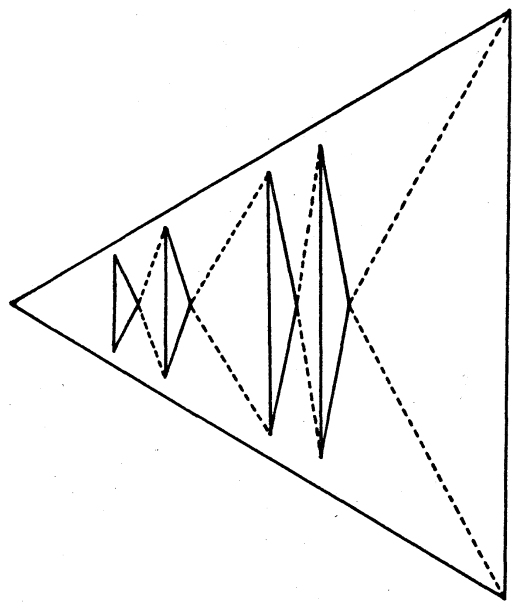

Fig. 2 Two visible vertices from each middle vertex of a window.

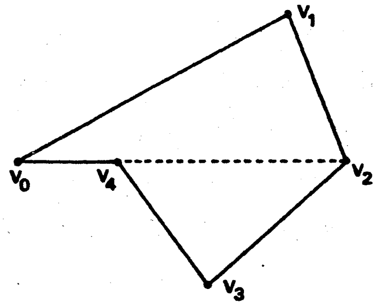

Fig. 3 Three collinear vertices.

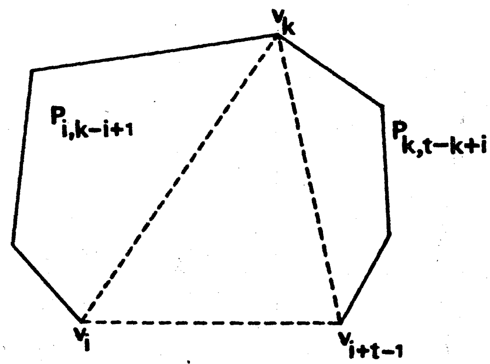

Fig. 4 Decomposition of a subpolygon $P_{i,t}$ into three parts: Two subpolygons $P_{i,k-i+1}$ and $P_{k,t-k+i}$ and a triangle (v_i, v_k, v_{i+t-1}).

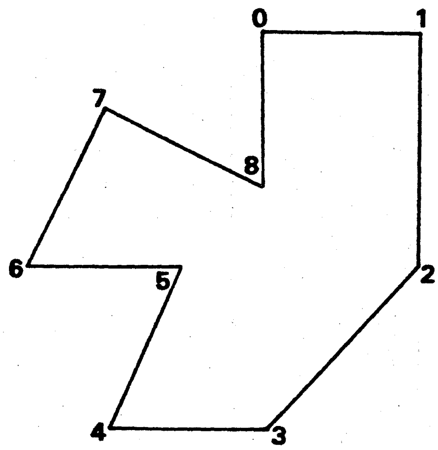

Fig. 5 A polygon to illustrate the algorithm.

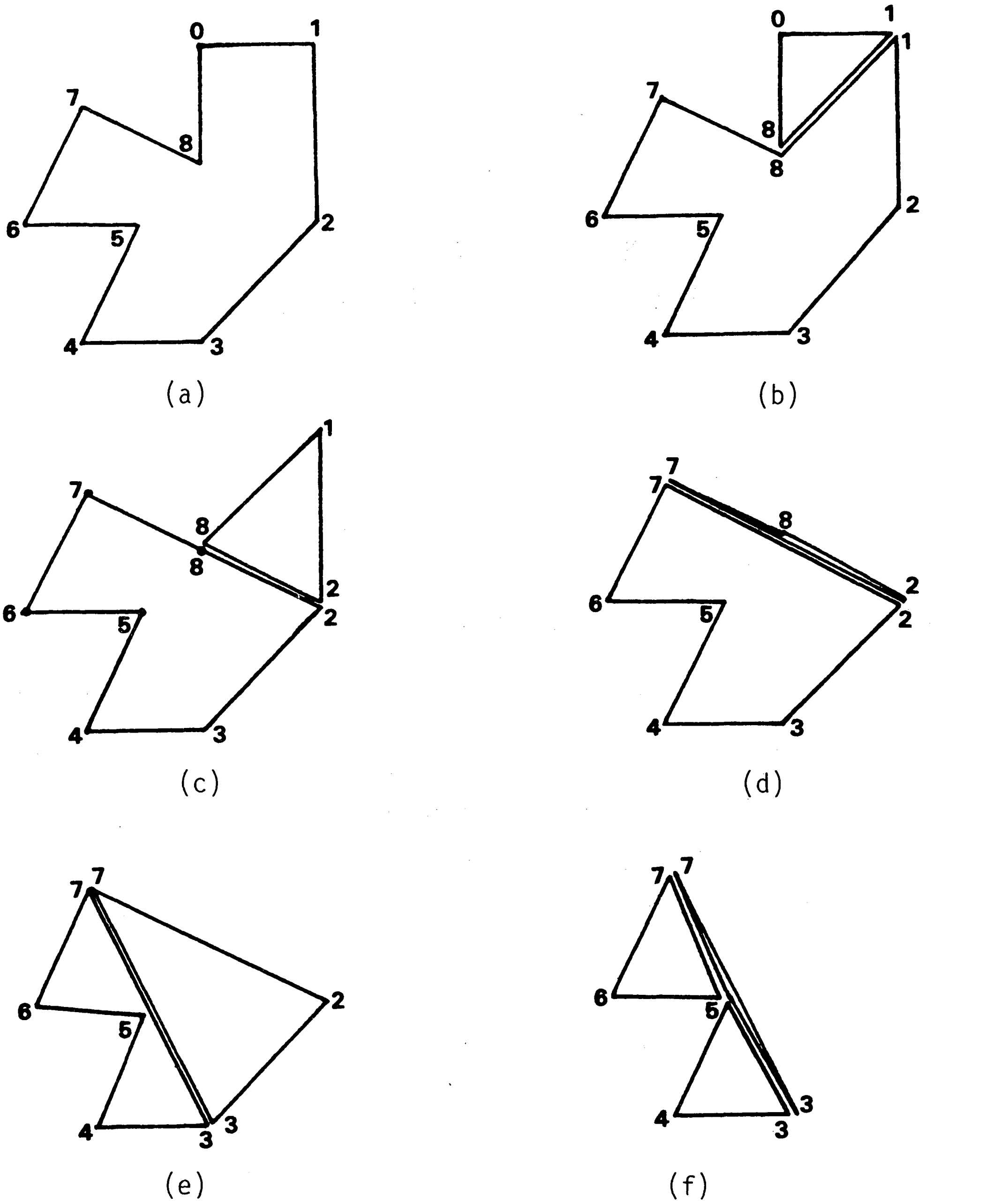

Fig. 6 Illustrative example. (a) original polygon, (b) decomposition of $P_{0,9}$ into $P_{1,8}$ and Tri(0,1,8), (c) decomposition of $P_{1,8}$ into $P_{2,7}$ and Tri(1,2,8), (d) decomposition of $P_{2,7}$ into $P_{2,6}$ and a dummy triangle (2,7,8), (e) decomposition of $P_{2,6}$ into $P_{3,5}$ and Tri(2,3,7), and (f) decomposition of $P_{3,5}$ into $P_{3,3}$, $P_{5,3}$, and a dummy triangle (3,5,7).