Advanced Science and Technology Letters
Vol.51 (CES-CUBE 2014), pp.207-210
http://dx.doi.org/10.14257/astl.2014.51.47

Linear-Time Algorithms for Finding Constant Visible Candidate Edges in a Polygon with Holes

Jong-Sung Ha[1] and Kwan-Hee Yoo[2]

[1]Department of Game and Contents, Woosuk University
490 Samryeup, Wanjugun, Chonbuk,, South Korea
jsha@woosuk.ac.kr
[2]Department of Software Engineering,Chungbuk National University,
52 NaesudongroHeungdukguCheongjusiChungbuk, South Korea
khyoo@chungbuk.ac.kr

Abstract. This paper presents concrete linear-time algorithms for finding the constant number of visible candidate edges in a polygon with holes. Our algorithms take the similar approach of Shin and Woo [1] in a polygon without holes.

Keywords: Computational geometry, Edge visibility, Gallery problem

1 Introduction

An edge in a polygon is called a *(weakly) visible edge* if every other point in the polygon is visible from some point in the edge. A polygon with holes, which will be called a *holed-polygon* for brevity, is a natural model in many geometric applications such as the gallery problem [4]; obstacles in a gallery are treated as holes within a polygon. Shin and Woo [1] have provided an $O(n)$ time algorithm for finding all visible edges in a polygon without holes, and another optimal algorithm has been presented by Sack and Suri [2]. However, this problem is still challenging in a holed-polygon; only a result on the number of visible candidate edges has been presented by Park *et al*. [3], but any efficient method is still not known for determining the visibility for each candidate except the $\Omega(n^4)$ time method [5] of constructing visibility polygons from the candidate edges in a holed polygon. This paper is concerned with developing concrete linear time algorithms for finding the visible candidate edges in a holed-polygon.

2 Determining Visible Candidate Edges in a Holed-Polygon

There are well-known linear-time algorithms [1,2] for finding all visible edges in a polygon without holes. In a polygon with holes, only a tight bound on the number of visible edges has been established; there may be at most three on the boundary and three on one of the holes [3].This section presents linear time algorithms for finding

ISSN: 2287-1233 ASTL

the constant number visible candidate edges in a holed polygon, by applying the similar approach of [1] in a polygon without holes into the theoretical results of [3].

First, we locate three distinct edgesin a convex hole$H$ such that $HP(v_i,v_{i+1}) \cap HP(v_j,v_{j+1}) \cap HP(v_k,v_{k+1}) = \emptyset$, where $HP(u,v)$denotesthehalfplanecontaining anedge$E(u,v)$suchthat everypoint zin$HP(u,v)$lieseitherontheline$L(u,v)$orliestotheleftof$E(u,v)$.Let $ME(v_i,v_{i+1})$ be the mapped edge of $E(v_i,v_{i+1})$ such that v_i is located at origin.

```
FIND-IJK(i, j, k)
begin
        k = 0
        j = 1
        while ME(vj, vj+1) is to the left of ME(vk+1, vk) do
                j = j + 1
        endwhile
        i = j − 1
        if ME(vj, vj+1) is on ME(vk+1, vk) then
                j = j + 1
        endif
end
```

Second, we find allcandidates ofouter visible edges. After assigningcolors“red”, “green”, and“yellow” toeachofthethreevisibilitypolygons defined with the middle points of the above three distinct edges, where the *visibility polygon*$V(q,P)$[6] isdefinedasthesetofallpointsin$P$whichisvisible from q, wefind all full-colored edgesof$P$; this problem is equivalent to finding allcandidates ofouter visible edgesand thenumberofall full-colored edges isatmostthree [3].

```
EDGE-CLR-OUT(r, Color)
begin
        mr = 1/2(vr+vr+1)
        compute V(mr, P)
        b = 0
        q = 0
        while q < b do
                while sq ∉ E(vb, vb+1) do
                        b = b + 1
                endwhile
                mark E(vb, vb+1) with Color
                if sq ∈ E(vb+1, vb+2) then
                        mark E(vb+1, vb+2) with Color
                endif
                q = q +1
        endwhile
end

FIND-CD-OUT(H, P)
begin
        call FIND-IJK(i, j, k)
        call EDGE-CLR-OUT(i, "red")
        call EDGE-CLR-OUT(j, "green")
        call EDGE-CLR-OUT(k, "yellow")
        CAND = ∅
        f = 0
        while f < n do
                if E(vf, vf+1) is full-colored then
                        CAND = CAND ∪ E(vf, vf+1)
                endif
                f = f +1
        endwhile
        return with CAND
end
```


Third, wepresentanotherbasicoperationfordeterminingcandidates ofinnervisible edgesinaholed-polygon.

```
EDGE-CLR-IN(E(u,v), Color)
begin
    for j = 0 to 2 do
        i = mod(j-1,3)
        k = mod(j+1,3)
        if v_2i is in HP(u,v) then
            mark E(v_2i ,v_2j) and E(v_2i v_2k) with Color
        endif
    endfor
end

FIND-CD-IN(T_1 , T_2)
begin
    call EDGE-CLR-IN(E(v_10, V_11), "red")
    call EDGE-CLR-IN(E(v_11, V_12), "green")
    call EDGE-CLR-IN(E(v_12, V_10), "yellow")
    CAND = φ
    for i = 0 to 2 do
        j = mod(i+1,3)
        if E(v_2i , v_2j ) is full-colored then
            CAND = CAND ∪ E(v_2i , v_2j )
        endif
    endfor
    return with CAND
end
```

Finally, wedescribe an algorithm tofindcandidate edgesin $[P;H_1,\ldots,H_h]$ where $h\geq 2$, usingthe previousfundamental operations.

```
FIND-CD([P;H_1, . . .,H_h])
begin
    i = 0
    CAND = ∅
    while i < h do
        C = CONVEX(H_i)
        if C is not triangular then
            i = i + 1
            continue
        endif
        if CAND = ∅ then
            WIN = C
            CAND = H_i ∩ C
        else
            CAND = CAND ∩ FIND-CD-IN(C, WIN)
            if CAND = ∅ then
                WIN = C
                CAND = H_i ∩ FIND-CD-IN(C, WIN)
            endif
        endif
        i = i + 1
    endwhile
    CAND = CAND ∪ FIND-CD-OUT(C, P)
end
```


3 Concluding Remarks

The coloring approach [1] edge visibility in a polygon without holes and the upper bound on number of visible edges [3] in a holed-polygon together provide us new penetrations to the visibility problem in a holed-polygon. Based on their results, we established algorithms for determining a constant number of all candidate edges in a polygon with one hole and multiple holes, respectively. These algorithms run in $O(n)$ time, where n is the number of vertices of all holes and vertices of the polygon boundary.

Acknowledgements. This research was partially supported by Woosuk University and by theMSIP(Ministry of Science, ICT and Future Planning), Korea, under theITRC(Information Technology Research Center) support program(NIPA-2014-H0301-14-1022) supervised by the NIPA(National IT Industry Promotion Agency).

References

1. Shin, S., and Woo,T.: An optimal algorithm for finding all visible edges in a simple polygon, IEEETr. on Robotics and Automation,5, 202-207 (1989).
2. Sack, J. andSuri,S.: An optimal algorithm for detecting weak visibility of a polygon, IEEE Tr. on Comput., 39, 1213-1219 (1992).
3. Park, J., Lee, D. and Lin,A.: Computational complexity of art gallery problems, IEEE Tr. on Inform.Theory, IT-32, 12, 276-282(1986).
4. Shin,S.,Chwa, K. and Woo, T.: On the number of guard edges of a polygon, Discrete& Computational Geometry, 10, 447-562 (1993).
5. Suri, S. and O'Rourke,J.: Worst-case optimal algorithms for constructing visibility polygons with holes, Proc. 2nd ACM Symp. Computational Geometry, 14-23(1986).

