

A Simple Polygon Search Algorithm

Sang-Un Lee*

Abstract

This paper considers simple polygon search problem. How many searchers find a mobile intruder that is arbitrarily faster than the searcher within polygon art gallery? This paper uses the visibility graph that is connected with edges for mutually visible vertices. Given visibility graph, we select vertex u that is conjunction $\Delta(G)$ in $N_G(v)$ for $d_G(v) \leq 4$. We decide 1-searchable if $1 \leq |u| \leq 2$ and 2-searchable if $|u| \geq 3$. We also present searcher's shortest path. This algorithm is verified by varies 1 or 2-searchable polygons.

► Keywords : Polygon search problem, Searcher, Intruder, Searcher's path, Flashlight search schedule

• First Author: Sang-Un Lee, Corresponding Author: Sang-Un Lee

*Sang-Un Lee (sulee@gwnu.ac.kr), Dept. of Multimedia Engineering, Gangneung-Wonju National University

• Received: 2016. 03. 19, Revised: 2016. 04. 04, Accepted: 2016. 04. 26.

I. Introduction

다각형 내부에 장애물을 갖고 있지 않는 다각형(polygon without holes)을 단순 다각형(simple polygon)이라 한다. 단순 다각형 P 로 구성된 화랑(art gallery)에 침입자(intruder, evader)가 있는 경우 몇 명의 수색자(searcher, pursuer)로 결국에는 침입자를 찾아낼 수 있는가? 이 문제를 Suzuki와 Yamasita[1]가 제시한 다각형 탐색 문제(polygon search problem, PSP)라 하며[2-6], 다음과 같이 가정한다.

- (1) 다각형 화랑 P 는 n 개의 모서리 (꼭지점, 정점)와 n 개의 벽 (간선) ∂P 로 닫힌 공간으로 해밀턴 사이클로 볼 수 있으며, 내부는 어두운 밤에 전등이 켜져 있지 않아 전혀 볼 수 없다.
- (2) 침입자와 수색자는 계속적으로 움직인다. 그러나 침입자의 이동 속도는 수색자보다 빠르다.
- (3) 수색자는 회중전등(flashlight)을 가지고 있으며, 회중전등으로 비출 수 있는 영역만 볼 수 있기 때문에 다각형의 벽 ∂P 를 따라 이동한다.
- (4) 침입자는 벽이나 내부 공간 어디에 존재하여도 상관없다. 침입자가 숨어있는 부분을 오염된(contaminated) 영역이라 하며, 그렇지 않으면 투명(clear)하다고 한다.
- (5) 회중전등이 비출 수 있는 수평 거리는 무제한으로 다각형 내부의 가장 긴 대각선 끝까지 가능하며, 수직으로는 천장에서 바닥까지 전체를 비춘다.
- (6) 만약, 침입자가 회중전등이 비추는 영역을 지나가거나, 회중전등이 침입자를 직접 비추거나 침입자가 수색자와 맞닥뜨리면(동일한 지점에 위치) 침입자를 체포한 것으로 간주한다.

이러한 가정 하에서, 우리는 침입자의 이동 속도와는 무관하게 결국에는 침입자를 체포할 수 있는 수색자 수와 최단 거리의 수색 경로를 찾고자 한다. 여기서 수색자가 이동하는 경로를 순찰 경로(patrol route, p -path), 회중전등이 비추는 경로를 탐조등 일정(searchlight schedule, f -path)이라 하자.

Suzuki와 Yamasita[1]는 다각형 내부를 탐색 가능한 필요충분조건(necessary or sufficient conditions)으로 k -searcher 또는 ∞ -searcher를 제시하였다. 여기서 ∞ -searcher는 수색자의 수가 ∞ 가 필요한 것이 아니라 1명의 수색자가 360° 를 볼 수 있는 경우이다. 이 문제에 대해 Tan[3], Lavalle et al.[4]과 Lee et al.[5]는 $k=1$ or 2 를 제시하였다. 즉 최대 2명으로 수색하면 결국에는 침입자를 체포할 수 있다. 다각형의 정점(꼭지점, 모서리) 개수를 $n, m \leq n$ 이라 할 때, Tan[3]과 Lee et al.[5]은 $O(n^2)$, Lavalle et al.[4]은 $O(m^2 + m \log n + n)$ 의 알고리즘을 제안하였다. 수색자의 최단 이동경로에 대해서는 가시도(visibility diagram, V-diagram)를 작성하여 구하는 방법이 연구되고 있다[4,6-9].

PSP와 유사한 개념으로 화랑문제(gallery problem, GP)가

있다. 화랑문제에서는 이동하는 수색자 개념이 아닌 고정된 위치의 경비원 수를 찾는 문제로, 고정된 위치에서 화랑의 모든 고정된 지점들을 경비할 수 있는 최소의 경비원수를 찾는 문제로 Lee[10]가 연구하였으며, 경비원이 이동하면서도 모든 지점들을 빠뜨리지 않고 경비할 수 있는 최단 경로를 찾는 방법으로 Lee와 Choi[11]가 연구하였다. PSP는 수색자와 침입자 모두 움직이는 경우로 GP와는 차별성이 있어 GP에 적용되는 알고리즘으로는 직접 적용이 불가하여 다른 방법을 고안해야 한다.

본 논문에서는 $k=1$ or 2 인 $O(n)$ 알고리즘을 제안한다. 2장에서는 다각형 탐색 문제의 예를 고찰해 본다. 3장에서는 $O(n)$ 복잡도를 가진 다각형 탐색 문제 알고리즘을 제안한다. 4장에서는 제안된 알고리즘의 적합성을 검증한다.

II. Polygon Search Problem

그림 1의 P_1 과 P_2 는 LaValle et al.[4]에서 인용된 사례이다. P_1 에 대해 초기에 수색자는 “0”에 위치하고 회중전등을 “0”에 비춘다. “0”에서 “14”로 이동하면서 회중전등을 “0”에서 “1”로 회전시킨다. “14”에 머무르면서 회중전등을 “1”에서 2,3,4를 거쳐 “5”로 회전시킨다. 다음으로 “5”를 계속 비추면서 “14”에서 “13”으로 이동한다. 그 다음에 회중전등을 “5”에서 6을 거쳐 “7”로 회전시킨다. 다음으로 회중전등을 “7”에 계속 비추면서 “13”에서 “10”으로 이동한다. “10” 위치에서 회중전등을 “7”에서 “8”로 회전시킨다. “10”에서 “9”로 이동하면서 회중전등을 “8”에서 “9”로 회전시키면 다각형 내부를 연속하여 모두 탐색하고 종료된다.

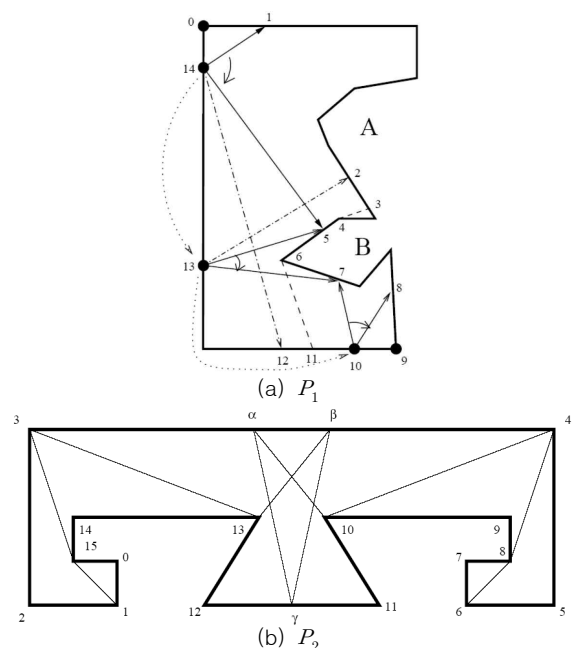


Fig. 1. Examples of Simple polygon search problem

결국, 수색자 이동 경로(p -path)는 “0-14-13-10-9”이며, 회중전등 탐색 일정(f -path)은 “0-2-3-4-5-6-7-8-9”로 1-searchable이다. LaValle et al.[4]은 이를 승리 경로(winning path)로 표현하였다. P_1 다각형의 승리경로는 $\langle 0,0 \rangle, \langle 14,1 \rangle, \langle 14,5 \rangle, \langle 13,5 \rangle, \langle 13,7 \rangle, \langle 10,7 \rangle, \langle 10,8 \rangle, \langle 9,8 \rangle, \langle 9,9 \rangle$ 이며, P_2 다각형은 $\langle 0,0 \rangle, \langle 15,1 \rangle, \langle 15,3 \rangle, \langle 13,3 \rangle, \langle 13,\beta \rangle, \langle \gamma,\beta \rangle, \langle \gamma,\alpha \rangle, \langle 10,\alpha \rangle, \langle 10,4 \rangle, \langle 8,4 \rangle, \langle 8,6 \rangle, \langle 7,7 \rangle$ 로 1-searchable이다. 이를 p -path와 f -path로 표현하면 표 1과 같다.

Table 1. p -path and f -path for P_1 and P_2

(a) P_1

	초기치	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7
p -path	0	0→14	14	14→13	13	13→10	10	10→9
f -path	0	0→1	1→2→3 →4→5	5	5→6→7	7	7→8	8→9

(b) P_2

	초기치	Step 1	Step 2	Step 3	Step 4	Step 5
p -path	0	0→15	15	15→14→13	13	13→12→ γ
f -path	0	0→1	1→2→3	3	3→ α → β	β

	Step 6	Step 7	Step 8	Step 9	Step 10	Step 11
p -path	γ	α →11→10	10	10→9→8	8	8→7
f -path	β → α	α	α → β →4	4	4→5→6	6→7

LaValle et al.[4]는 수색자 이동 경로를 행으로, 회중전등 탐색 경로를 열로 하는 2차원 그리드(grid)의 가시성 장애물도(visibility obstruction diagram)를 그려 $O(n^2)$ 알고리즘을 제안하였다. Lee et al.[5]은 $2n \times 2n$ 그리드를 작성하여 $O(n^2)$ 알고리즘을 제안하였다. 3장에서는 $O(n)$ 알고리즘을 제안한다.

III. Linear Polygon Search Algorithm

본 장에 적용되는 그래프 알고리즘 개념은 다음과 같다. 그래프 $G=(V,E)$ 에서 정점 v 의 차수(부속 간선 수)를 $d_G(v)$ 라 한다. 그래프의 최소 차수 정점을 $\delta(G)$, 최대 차수 정점을 $\Delta(G)$ 라 하며, 이웃하는 정점을 $N_G(v)$ 라 한다. 다각형 P 는 n 개의 정점(꼭지점)과 n 개의 간선(벽)으로 연결된 닫힌 해밀턴 사이클로 기본적으로 각 정점의 차수는 $d_G(v_i)=2$ 이다. 다각형 탐색 알고리즘은 다음과 같이 수행된다.

- (1) 만약, 내각<90°인 연속된 두 정점 존재시, 두 점점의 변을 다각형 내부로 연장한 선분이 만나는 점에 정점을 추가한다. 예로, 그림 1의 (b)에서 ⑫를 볼 수 있는 정점은 β , ⑪은 α 이다.
- (2) 각 정점 u 에서 대각선으로 볼 수 있는 모든 다른 정점 v 로 간선을 연결한 가시성 그래프 G_s 를 그린다. $\delta(G)$ 인 정점 u 의 $N_G(u)$ 중 $\Delta(G)$ 정점을 s 로 “1”을 부여하고, 시계방향으로 연속적으로 번호를 부여한다.
- (3) 내각<90°인 연속된 2개 정점을 제외한 $d_G(v) \leq 4$ 인 정점들 v 를 선택하고 ∂P 간선을 연결한다. 필요한 수색자의 위치 u 는 다음과 같이 결정한다.
 - 만약, v_1 이 독립정점이고 다른 독립정점 v_2 와의 교집

합이 존재하지 않으면 식 (1)과 같이 $N_G(v_1)$ 들 중 $\Delta(G)$ 인 정점 u 를 선택한다.

if v_1 and v_2 독립정점 and $v_1 \cap v_2 = \{\phi\}$ then

$$u = \Delta(G) \in N_G(v_1) \tag{1}$$

- 만약, 두 독립정점 v_1 과 v_2 의 $u = N_G(v_1) \cap N_G(v_2)$ 의 $\Delta(G)$ 가 존재한다면 식 (2)와 같이 u 를 선택한다.

if v_1 and v_2 독립정점 and $u = N_G(v_1) \cap N_G(v_2)$ and $u = \Delta(G)$ then u 선택 $\tag{2}$

- 만약, v_1 이 연결된 정점들 v_1, v_2, v_3, \dots 이면 식 (3)과 같이 $u = N_G(v_1) \cap N_G(v_2) \cap N_G(v_3) \cap \dots$ 인 $\Delta(G)$ u 정점을 선택한다.

if $v_1 = \text{연결된 정점들 } v_1, v_2, v_3, \dots$ and $u = N_G(v_1) \cap N_G(v_2) \cap N_G(v_3) \cap \dots$ and $u = \Delta(G)$ then u 선택 $\tag{3}$

이들 정점의 개수를 $|u|$ 라 하면, 식 (4)와 같이 u 의 개수 $|u|$ 로 경비원(수색자) 수를 결정한다.

if $1 \leq |u| \leq 2$ then 1-searchable $\tag{4}$
 else if $|u| \geq 3$ then 2-searchable.

- (4) u 정점들을 대상으로 s 에서 최단 거리 정점을 t 로 결정한다. (s,t) 의 최단 거리를 수색자 #1의 이동 경로(s -path)로, 반대편 방향을 회중전등 탐색경로(f -path)로 설정한다.

- p -path 행과 $s+(f\text{-path})+t$ 열의 인접행렬(adjacency matrix)에서 가시성 간선이 존재하면 “1”, 그렇지 않으면 “0”으로 표기한다. $(s,s)=1, (t,t)=1$ 이다.
- 1-searchable 탐색 방법
수색자 #1은 행의 (s,t) 의 p -path의 i 번째 정점 u 의 “1”의 연속이 끝나는 정점 w 과 $i+1$ 번째 정점 v 의 “1”의 연속이 시작되는 정점 w 을 연결한 f -path를 탐색한다. 이때 추가된 정점은 “Z” 형태로 움직이고 간주한다.
- 2-searchable 탐색 방법
(1) s -path의 i 번째와 $i+1$ 번째 정점의 f -path 교차 정점 w 가 없으면 u 의 다른 정점(w 의 ∂P 인접 정점이 아닌 대각선에 존재하는 $\Delta(G)$ 정점)에 s_2 (수색자 #2)가 위치하며 w 에 회중전등을 비춘다.
(2) 또 다른 방법은 $\delta(G)$ 인 정점 u 의 $N_G(u)$ 중 시계방향과 반시계방향의 $\Delta(G)$ 정점에 각각 수색자 #1과 수색자 #2가 배치되어 상호 볼 수 없는 영역을 보완해가면서 탐색할 수 있다.

P_1 에 제안 알고리즘을 적용하면 그림 2와 같다. $d_G(v) \leq 4$ 인 정점 v 를 “O”로, 연속된 $N_G(v)$ 들의 교집합인 $\Delta(G)$ 인 정점을 “□”로 표기한다. 가시성 그래프에서 $\delta(G)$ 인 정점은 ②와 ⑪이다. 임의로 ②를 선택하고 $N_G(2)$ 인 ①과 ③ 중에서 $\Delta(G)$ 는 ①로 $s \leftarrow 1$ 로 설정한다. 다음으로 $d_G(v) \leq 4$ 인 정점들 2-3-4-5-6-7-8과 10-11-12에 대해 교집합을 구하면 2-

동하여도 ⑥을 탐색할 수 없다. 따라서 수색자 #2는 “15”의 $N_G(15)$ 의 $\Delta(G)$ 인 “16”에 배치되어야 하나 수색자 #1이 (1,2) 경로의 ①에서 최종적으로 볼 수 있는 정점이 “17”이기 때문에 “17”을 볼 수 있는 ⑬에 위치한다. 이후에는 2명의 수색자가 빈 공간을 허용하지 않으면서 탐색한다.

IV. Applications and Evaluation

본 장에서는 그림 5의 $1 \leq |u| \leq 2$ 인 1-searchable과 그림 6의 $|u| \geq 3$ 인 2-searchable 다각형에 대해 제안된 알고리즘을 적용하여 본다.

P_4 는 Wikipedia[12]에서, P_5 은 Do[13]에서, P_6 은 Zhang과 Kameda[6], Bahun과 Libiw[8], Bhattacharya et al.[9]에서, P_7 은 Urrutia[14]에서, $P_8, P_{12}, P_{13}, P_{14}, P_{15}, P_{16}$ 은 Yamashita et al.[2]에서, P_9 은 Peterson[15]에서, P_{10} 은 Ghosh[16]에서, P_{11} 은 Shermer [17]에서, P_{17} 은 Isler et al.[18]에서, P_{18} 은 Kameda et al.[7]에서 인용되었다.

1-searchable 다각형에 제안된 알고리즘을 적용한 결과는 그림 5에, 2-searchable 다각형에 제안된 알고리즘을 적용한 결과는 그림 6에 제시되어 있다.

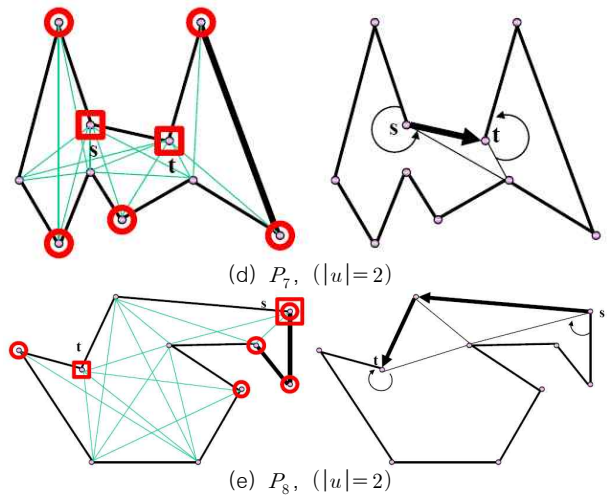
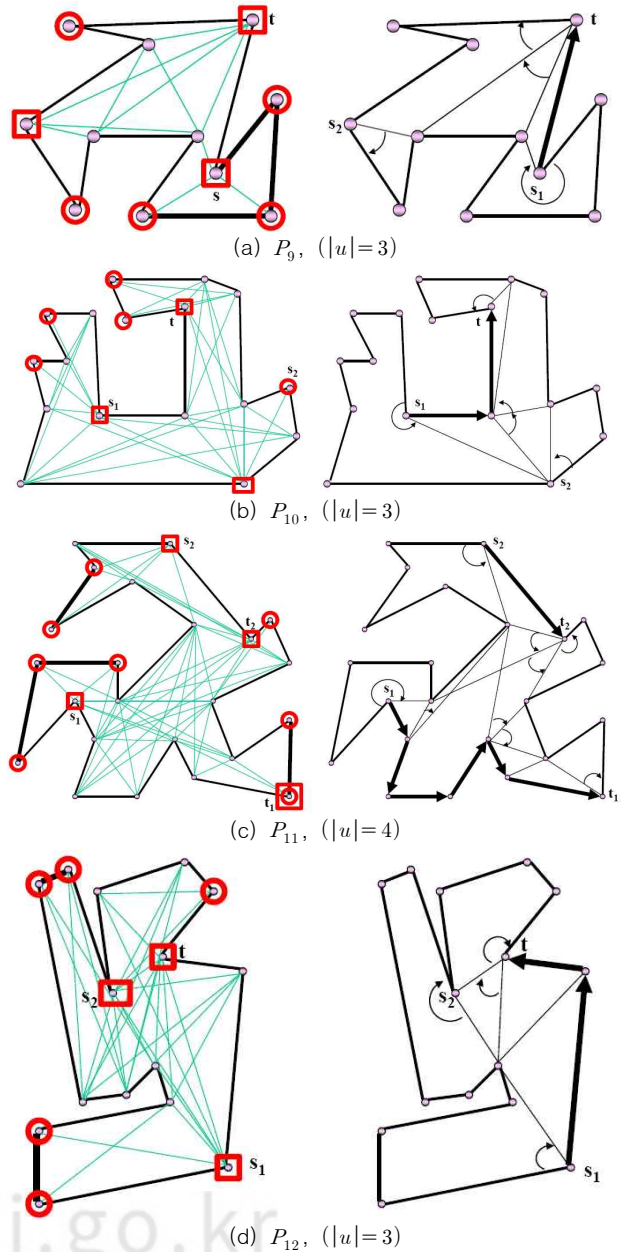
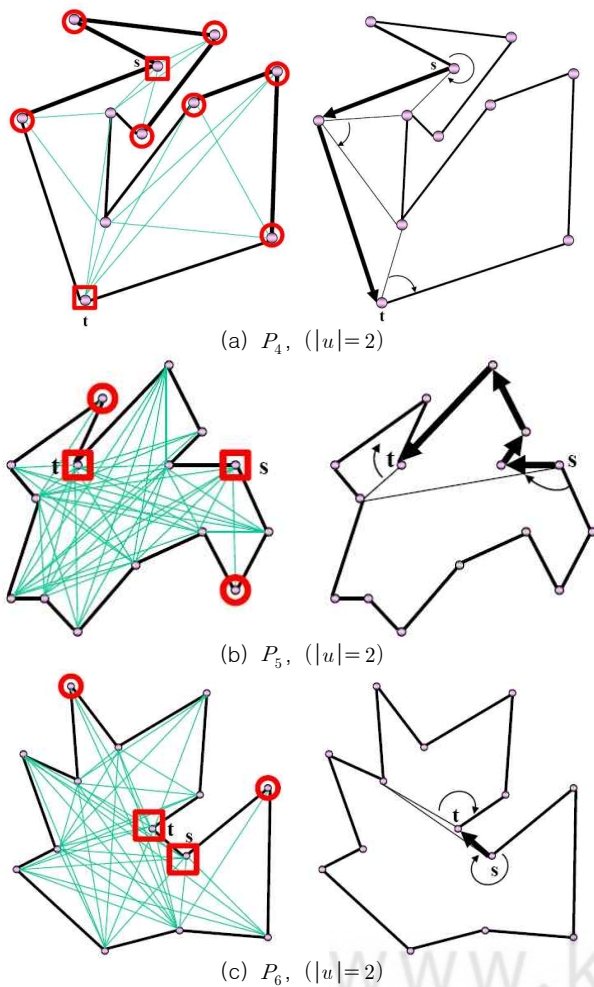
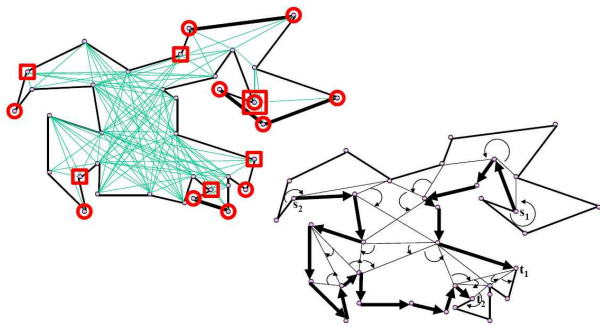
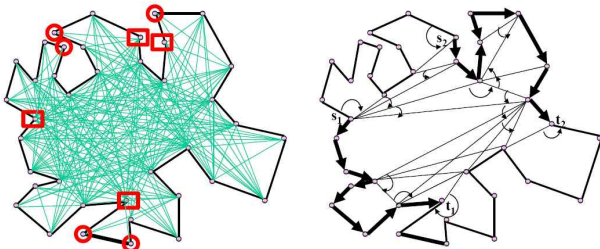


Fig. 5. Proposed algorithm for 1-searchable polygons

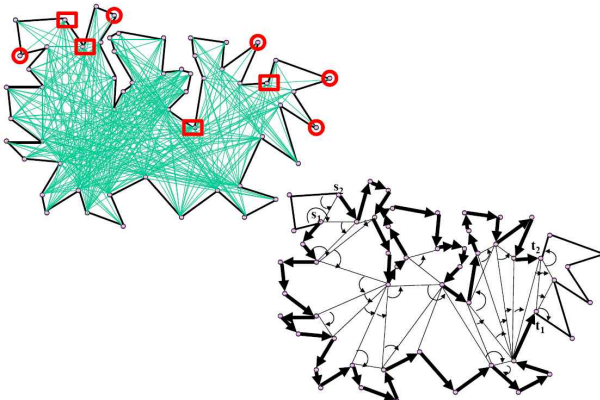




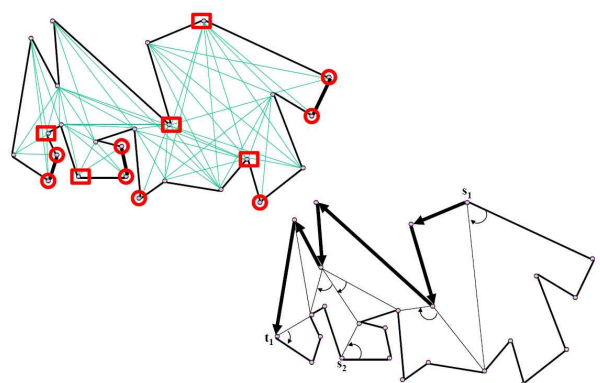
(e) P_{13} , ($|u|=6$)



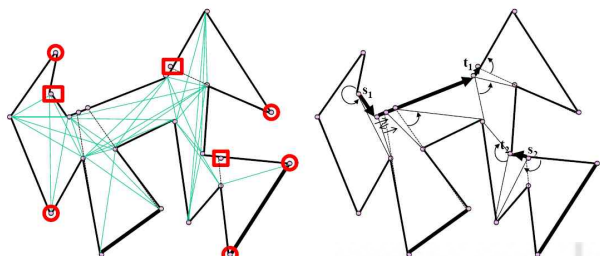
(f) P_{14} , ($|u|=4$)



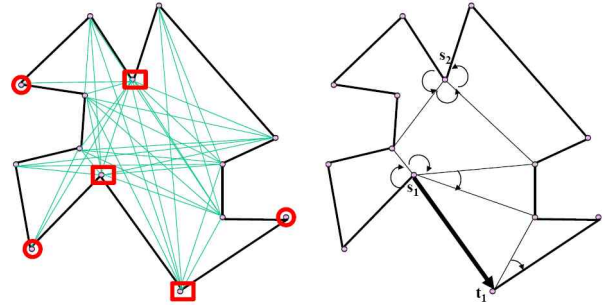
(g) P_{15} , ($|u|=4$)



(h) P_{16} , ($|u|=5$)



(i) P_{17} , ($|u|=3$)



(j) P_{18} , ($|u|=3$)

Fig. 6. Proposed algorithm for 2-searchable polygons

제안된 알고리즘을 그림 5의 1-searchable 다각형과 그림 6의 2-searchable 다각형에 적용한 결과 $k=1$ or 2임을 알 수 있으며, 동시에 수색자의 최소 이동경로도 보여주고 있다. 결국, 제안된 알고리즘은 k 값을 쉽게 구할 뿐 아니라 수색자의 최소 이동경로(p -path)도 동시에 제공해주는 장점을 갖고 있다.

V. Conclusions

본 논문은 단순 다각형 화랑에 침입자가 들어왔을 때 결국 몇 명의 수색자가 필요하며 어떤 경로로 탐색해야 최적인지를 구하는 단순 다각형 탐색 문제를 고찰하였다. 제안된 알고리즘은 모든 정점(꼭지점)에서 볼 수 있는 대응되는 정점 간에 간선을 그린 가시성 그래프를 그린다. 이 가시성 그래프에서 $d_G(v) \leq 4$ 인 정점 v 를 선택한다. 연속된 $N_G(v)$ 들의 교집합인 $\Delta(G)$ 인 정점 u 를 선택한다. 만약, $1 \leq |u| \leq 2$ 면 1-searchable, $|u| \geq 3$ 면 2-searchable로 결정하였으며, u 정점 들 간의 최단 경로를 수색자 이동 경로 (p -path)로 결정하였다. 또한, 한 정점에서 최대한으로 볼 수 있는 정점까지 회중전등을 비추고 다음 정점으로 이동하는 방법을 택하였다. 제안된 알고리즘을 다양한 1 또는 2-searchable 다각형에 적용하여 알고리즘 적합성을 검증하였다.

REFERENCES

- [1] I. Suzuki, and M. Yamasita, "Searching for a Mobile Intruder in a Polygon Region," SIAM Journal on Computing, Vol. 21, No. 5, pp. 863-888, 1992.
- [2] M. Yamashita, H. Umemoto, I. Suzuki, and T. Kameda, "Searching for Mobile Intruders in a Polygonal Region by a Group of Mobile Searchers," SIAM Journal on Computing, Vol. 21, pp. 863-888, 1996.
- [3] X. Tan, "Searching a Simple Polygon by a k -Searcher," Lecture Notes in Computer Science (LNCS), Vol. 1969, pp. 503-514, 2000.
- [4] S. M. LaValle, B. Simov, and G. Slutzki, "An

- Algorithm for Searching a Polygonal Region with a Flashlight," *International Journal of Computational Geometry and Applications (IJCGA)*, pp. 260-269, 2000.
- [5] J. H. Lee, S. M. Park, and K. Y. Chwa, "Simple Algorithms for Searching a Polygon with Flashlights," *Information Processing Letters*, Vol. 81, pp. 265-270, 2002.
- [6] J. Z. Zhang and T. Kameda, "Where to Build a Door," Technical Report, CMPT2006-14, Department of Math and Computer Science, University of Lethbridge, Canada, 2006.
- [7] T. Kameda, I. Suzuki, and J. Z. Zhang, "Minimization of Distance Traveled in Surveillance of a Polygonal Region from the Boundary," *Japan Conference on Computational Geometry and Graphs (JCCGG)*, 2009.
- [8] S. Bahun and A. Lubiw, "Optimal Schedules for 2-Guard Room Search," *CCCG*, 2007.
- [9] B. Bhattacharya, J. Z. Zhang, Q. Shi, and T. Kameda, "An Optimal Solution to Room Search Problem," Technical Report, CMPT2006-16, School of Computing Science, Simon Fraser University, Canada, 2006.
- [10] S. U. Lee, "Minimum number of Vertex Guards Algorithm for Art Gallery Problem," *Journal of KSCI*, Vol. 16, No. 6, pp. 179-186, Jun. 2011.
- [11] S. U. Lee and M. B. Choi, "The Minimum number of Mobile Guards Algorithm for Art Gallery Problem," *Journal of IIBC*, Vol. 12, No. 3, pp. 63-69, Jun. 2012.
- [12] Wikipedia, "Art Gallery Problem," http://en.wikipedia.org/wiki/Art_gallery_problem, 2016.
- [13] N. Do, "Art Gallery Theorems," www.austms.org.au/Publ/Gazette/2004/Nov04/mathellaneous.pdf, 2004.
- [14] J. Urrutia, "Art Gallery and Illumination Problems," *Handbook of Computational Geometry*, Department of Computer Science, University of Ottawa, Canada, 1996.
- [15] I. Peterson, "Problem at the Art Gallery," http://www.maa.org/mathland/mathland_11_4.html, 1996.
- [16] S. K. Ghosh, "Art Gallery Theorems and Approximation Algorithms," School of Technology & Computer Science, Teta Institute of Fundamental Research, Mumbai, India, 2010.
- [17] T. C. Shermer, "Recent Results in Art Galleries," *Proceedings of the IEEE*, Vol. 80, No. 9, 1992.
- [18] V. Isler, S. Kannan, and S. Khanna, "Locating and

Capturing an Evader in a Polygonal Environment," *In Proceedings of the Sixth International Workshop on Algorithmic Foundations of Robotics (WAFR'04)*, pp. 351-367, 2004.

Authors



Sang Un Lee received the B. Sc. degree in avionics from the Korea Aerospace University in 1997. He received the M. Sc. and Ph. D. degrees in Computer Science from Gyeongsang National University, Korea, in 1997 and 2001, respectively. He is currently Professor with the Department of Multimedia Science, Gangneung-Wonju National University, Korea. He is interested in software quality assurance and reliability modeling, software engineering, software project management, neural networks, and algorithm.