Finding the Visibility Matrix of an Orthogonal Polygon

Amrita Agarwala
Information Technology Department,
Bengal Engineering & Science University, Shibpur, Howrah, India
amrita.agarwala@gmail.com

Abstract. An algorithm to find the visibility matrix of an orthogonalpolygon is presented here. The algorithm is applied on the vertices of theinput polygon in an anti-clockwise manner to find the visibility matrixwhich indicates the visibility of other vertices from the given vertex. Thealgorithm uses combinatorial techniques to determine the visibility of avertex from a given vertex. The visibility matrix captures the visibility ofvertices from each vertex of an input polygon in the form of a matrix. Theruntime of the algorithm is $O(n^2)$. An analysis of the visibility matricesof different isothetic polygons derived from the different shape images.

1 INTRODUCTON

The visibility problem is an important research topic in computational geometry.For any two points *x* and *y* in the plane or space, *x* is said tobe visible from *y* or vice versa if and only if the line segment joining them doesnot intersect any object.Linear time algorithms for the problem of computing the visibility polygon are presented in [4] and [5]. Here, each edge of the polygon is assumed to beopaque. However, a modified version [8] of Lee's algorithm [4] is presented withthe proof of correctness. A linear time algorithm [9] deals with the similar problem for an orthogonal polygon.

The notion of visibility of the polygon from a given edge is presented in [10]. The problem is solved in o(nlog log n) time presented in [11].

In the three-dimensional case, the well-known visibility problem for a set ofpolyhedra is the removal of all edges or parts of edges that are hidden from anobserver at some position (viewpoint), and it is referred to as the hidden-lineelimination problem. The visibility problems for orthogonal objects in two-or three dimensions in o(n log n+k) time and o(n) space has been addressed in[12].There are various notions of visibility studied by researchers. In [13] clear visibility is introduced. Two points u and v in a polygon P are called clearly visible ifthe open line segment joining u and v lies in the interior of P.

Staircase visibility has been studied in [14], [15], [16]. If a path inside arectilinear polygon P is monotone with respect to both axes, the path is calledstaircase path in P. Two points u and v in Pare called staircase visible if thereis a staircase path between u and v in P.Rectangular visibility has beenpresented in [17], [18].Circular visibility, another variation of visibility, has been given in [19], [20].In[21] the study of X-ray visibility, which is another variation of visibility is given.Two points u and v are X-ray visible in a polygon P
if the segment uv doesintersect more than a fixed number of edges of P.Point visibility is dealt in [22].

In this paper, an algorithm to find the visibility matrix of an orthogonalpolygon is presented. The rest of the paper is organized as follows. The preliminary definitions are presented in Sec. 2. Section 3 presents the algorithm witha discussion on the time complexity. The experimental results on different inner polygons of different images are presented in Sec. 4. Section 5 presents theconclusion with a note on future direction of this work.

2 PRELIMINARIES

Definition 1: A subset of Z^2 in which every pair of points is k-connected, iscalled a k-connected set. A digital object A is said to be 8-connected subset of Z^2 whose complement $Z^2 \setminus S$ is a 4-connected set [23].

Definition 2: The background grid is given by G=(H,V), where H & V represent two sets of equispaced horizontal and vertical grid lines respectively. Thegrid size g is defined as the distance between two consecutivehorizontal/verticalgrid lines. A grid point is the point of intersection of a horizontal and a verticalgrid line [24].

Definition 3: The inner (isothetic) cover (IIC) [24], denoted by $P(S)$, isa set of inner polygons and (inner) hole polygons, such that the region, given bythe union of the inner polygons minus the union of the hole polygons, containsa unit grid block (UGB) if and only if it is a subset of S. The border BP of Pis the set of points belonging to its sides. The interior of P is the set of pointsin the union of its constituting UGBs excluding the border of P .An inner isothetic polygon P can be defined as follows:

Inner polygon:P∩ $S \neq \theta, for\ each\ p \in Bp, \exists q \in S' \setminus P\ s.t. 0 < d_T(p,q) \leq g.$

Definition 4: P is an orthogonal polygon if and only if each of its verticesis a grid point and each of its edges is axis-parallel.

Definition 5: The visibility graph, $G \in (V, E)$ of a simple polygon is defined asfollows. The vertices of the graph are the vertices of the polygon and $(v_i, v_j) \in E$ ifthe line segment joining the corresponding vertices in the polygon lies completelyinside the polygon.

Definition 6: The visibility matrix is a n x nmatrix for a polygon with n vertices defined as:
v[i,j]=1 ,if i is visible from j
0 ,otherwise

2.1 Deriving the Inner Isothetic Cover:

Inner isothetic cover(A_{in})of a digital object A with grid G is the maximum area orthogonal polygon that can inscribe into the object A. The algorithm TIPS [25] computes the ordered set of vertices of A_{in}using a combinatorial technique based on the fact that the grid points lying on/ inside/ outside the object boundary. A grid point p is classified into 5 categories based on how many of the four cells, each of size g x g, incident at p, are fully occupied by the object points (i.e.,pixels from A). Say, the number of fully occupied cells incident at q is $i \in [0,4]$. Then q is classifed to class $C_i\{i \in [0,4]\}$, as shown in Fig. 1. The classification of the classes are shown below.

(i) C_0:None of the 4 cells occupied by object point. So, q is not a vertex of A_{in};

(ii) C_1: Exactly 1 cell is occupied. q is a 90^0 vertex of A_{in} (Fig. 1(a));

(iii) C_2: (a) If two adjacent cells are fully occupied, then q is an edge point(Fig. 1(c));

(b) If diagonally opposite cells are fully occupied, then q is a 90^0 vertex of A_{in}
(Fig. 1(d));

(iv) C_3: q is classified as a 270^0 vertex (Fig. 1(b));

(v) C_4: q is not a vertex of $A_{in}$and lies inside A_{in} .

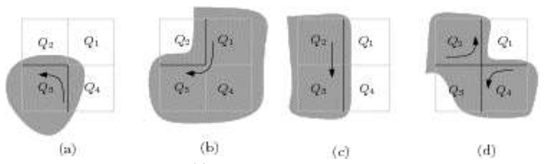


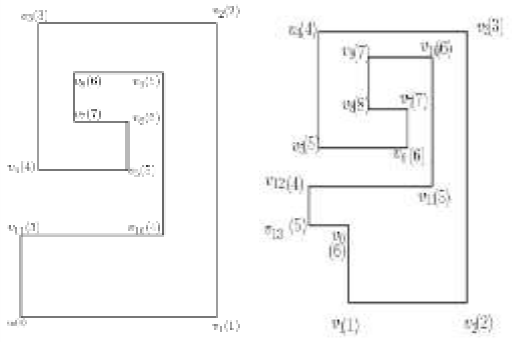
Fig1: Different vertex types

3 PROPOSED ALGORITHM

3.1 Algorithm

The visibility matrix of a given polygon is computed by traversing the polygon in an anticlockwise manner from the given vertex from which the visibilities of other vertices are to be computed. The same procedure is followed from othervertices to compute the complete visibility matrix. Let v_0 be the reference pointfrom which the visibility of other vertices are to becomputed. If v_1 is the vertexnext to v_0 in P, then v_0v_1 defines the reference axis. Each vertex v_i is associated with a coefficient of concavity, α_i, its relative angle μ_i with the reference axis(the angle between v_0v_1 and the edge v_0v_i). The concept of concavity coefficientis introduced to capture the fact when the concavity coefficient exceeds a specificvalue then the vertex is well inside a concavity, thus not visible from v_0. The internal angle of a vertex v_i is denoted by ϕ_i (either90^0or $270^0$for an isotheticpolygon). x_i, y_i are defined as the relative, unsigned distance between (v_0, v_1)along reference X_i axis and Y_i axis respectively having v_0 as origin and v_0v_1 as X_i reference axis. Also, the relative direction of traversal, either clockwiseoranticlockwise, from v_{i-1} to v_i is associated with v_i as d_i . A stack, S, is usedwhich contains the vertices visible from v_0 at a given point of traversal. As thetraversal proceeds to the next vertex, the visibility of the current vertex and theearlier vertices in the stack are analyzed using a combinatorial technique basedon α_i, ϕ_i ,and the distance of v_i from v_0. It may so happen that some of thevertices in S have to be popped. Once the traversal is completed, S contains theset of vertices which are visible from v_0 .The reference vertex, v_0 , can be either a90^0vertex, or a 270^0vertex. It may be noted that for the first vertex v_1, α_1 is 1. The value of α is incremented(decremented) by one for any anticlockwise (clockwise) movement during thetraversal of the polygon. It may also be notedthat when v_0 is a 90^0vertex, theconcluding vertex of the traversal, i.e., v_0, has α = 4.For example, in Fig. 2(a), each vertex with their concavity coefficients areshown. For the vertex v_6 (Fig. 2(a)), α_6 is 6, so v_6 is invisible from v_0. It is alsoevident that v_7 and v_8 having α_7 = 7 and α_8 = 6 are also invisible from v_0 .Similarly, if v_0 is a 270^0vertex, the concluding vertex of the traversal, i.e., v_0 ,has α = 6 as shown in Fig. 2 (b). Vertices $v_7; v_8; v_9$ having α_k = 7; 8; 7 with(k = 7; 8; 9 respectively) are not visible from v_0 .

Condition 1: For each point v_i , first α_i is checked whether it is greater than 5 (7) when v_0 is a 90^0(270^0) vertex, then it is concluded readily that v_i is not visible. Also v_{i-1} is popped in this case if it is in S.

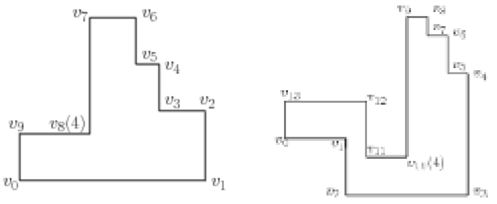


(a) (b)
Fig2: Example (a) 90^0(b)270^0vertex

Condition 2: Next, we check if$0^0 \le \theta_i \le 90^0$when v_0 is of type 90^0or if$0^0 \le \theta_i \le 270^0$when$v_0$ is of type 270^0then v_i is a candidate visible vertex.
Let v_{top} be the vertex at the top of stack S.

Condition 2a:When$\theta_i < \theta_{top}$, the following cases may occur.

A. v_0 is 90^0vertex:
Case 1: If $\theta_i < \theta_{top}$and $\alpha_i \le 3$, then v_i is discarded.
Case 2: If $\theta_i < \theta_{top}$and α_i = 4, then v_{i-1} is popped if in S.If $\theta_i < \theta_{top}$and$x_{top} > x_i$then v_{top} is popped. The popping of the top of S is continued till this condition issatisfied. If $0^0 < \theta_i < 90^0$, then v_i is visible (Case 2(a))(Fig. 3(a)) else v_i isnot visible (Case 2(b)) (Fig. 3(b)).



(a) (b)
Fig3: (a)Case2(a)(b) Case 2(b) (for90^0vertex)

Case 3: For α_i = 5, if θ_i is other than90^0, then it is not visible. v_{i-1} ispopped if it is in S. If $\theta_i < \theta_{top}$and $y_{top} > y_i$ then v_{top} is popped. The popping of the top of S is continued till this condition is satisfied.In Fig. 3(a), v_1 is initially pushed to stack, hence S = v_1. Next v_2 is pushed as $\theta_2 > \theta_{top}$ giving S = $v_1 v_2$. Similarly, v_k (k = 3, 4, 5, 6, 7) are pushed as $\theta_k > \theta_{top}$ (S = $v_1 v_2 v_3 v_4 v_5 v_6 v_7$). At v_8, $\theta_8 < \theta_{top}$(v_7 is currently the v_{top}). v_7 is popped as $\theta_8 < \theta_{top}$ and $y_{top} > y_8$. Similarly, v_6, v_5, v_4, v_3 are popped. v_8 is pushed as $0^0 < \theta_8 < 90^0$(Case 2(a)). In Fig. 3(b), v_1 to v_9 are processed in similar fashion,the obtained stack is S = $v_1 v_4 v_5 v_6 v_7 v_8 v_9$. When v_{10} isreached, as $\theta_{10} < 0^0$, itis not pushed to S (Case 2(b)). Whereas v_k (k = 9,8,7,6,5,4) are popped as per satisfiability of conditions for popping.In Fig. 4, when the traversal reaches v_5, the stack is having the vertices,S = $v_1 v_2 v_3 v_4$. As α_5 = 5 (Case 3) it is not pushed. However, v_4 is popped(S = $v_1 v_2 v_3$). Also $v_3$andv_2 are popped as the conditions $\theta_i < \theta_{top}$and $y_{top} > y_i$ are satisfied, giving S = v_1. Similarly, v_7 is discarded.

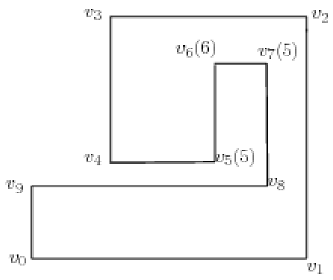
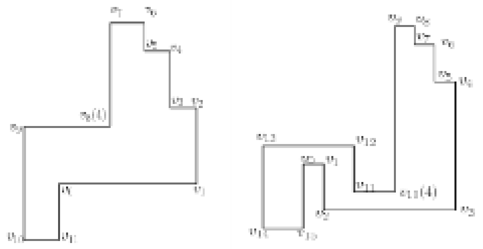


Fig4: Case3 (for 90^0 vertex)

B. v_0 is 270^0vertex:
Case 1: If $\theta_i < \theta_{top}$ and $\alpha_i \le 3$, then v_i is discarded.
Case 2: If $\theta_i < \theta_{top}$ and $\alpha_i = 4$, then v_{i-1} is popped if in S. If $\theta_i < \theta_{top}$ and $x_{top} > x_i$ then v_{top} is popped.The popping of the top of S is continued till this condition is satisfied. If $0^0 < \theta_i < 90^0$,

then v_i is visible (Case 2(a))(Fig. 5(a)) else v_i is not visible(Case 2(b))(Fig. 5(b)).



(a) (b)

Fig5: (a) Case2(a) (b) Case 2(b)(for 270^0vertex)

Case 3: If θ$\theta_i < \theta_{top}$and $\alpha_i = 5$ or 6, then v_{i-1} is popped if in S. If $\alpha_i = 5$, $\theta_i < \theta_{top}$and $y_{top} > y_i$ then v_{top} is popped. If $\alpha_i = 6$, $\theta_i < \theta_{top}$ and $x_{top} > x_i$ then v_{top} is popped.The popping of the top of S is continued till these conditionsare satisfied. v_i ispushed to S.

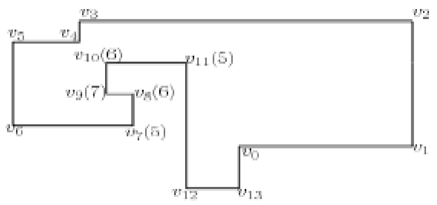


Fig6: Case3 and Case4 (for 270^0 vertex)

Case 4: If $\theta_i < \theta_{top}$ and $\alpha_i = 7$ then if θ_i is other than 270^0, then it is notvisible. v_{i-1} is popped if in S. If $\theta_i < \theta_{top}$ and $y_{top} > y_i$ then v_{top} is popped. The popping of the top of S is continued till this condition is satisfied.In Fig. 5(a), v_1 is initially pushed to S $= v_1$. Next v_2 is pushed as $\theta_2 > \theta_{top}$ giving S = $v_1 v_2$. Similarly, v_k (k = 3; 4; 5; 6; 7) are pushed to the stack givingS $= v_1 v_2 v_3 v_4 v_5 v_6 v_7$. At $v_8, \theta_8 < \theta_{top}$. v_7 is popped as $\theta_8 < \theta_{top}$ and $y_8 < y_{top}$. Similarly, v_6, v_5, v_4 are popped resulting S $= v_1 v_2 v_3$. v_8 is pushed as $0^0 \le \theta_8 \le 90^0$ giving S=$v_1 v_2 v_3 v_8$ at that point of traversal. In Fig. 5(b), v_1 to v_9 are processed in similar fashion and pushed to stack giving S = $v_1 v_4 v_5 v_6 v_7 v_8 v_9$.When we reach v_{10}, the vertex is having $\theta_{10} < 0^0$ hence not pushed to S. Whereas v_k (k = 9; 8; 7; 6; 5; 4) are popped resulting S = v_1 at that point of traversal.

In Fig. 6, at the end of traversal of v_6 the stack contains S=$v_1 v_2 v_3 v_4 v_5 v_6$. Next v_6 is popped as $\theta_7 < \theta_{top}$ and α_7 = 5 and v_7 is pushed (Case 3) givingS $= v_1 v_2 v_3 v_4 v_5 v_7$. Again v_7 is popped when traversal reaches v_8 (Case 3) with S $= v_1 v_2 v_3 v_4 v_5$. v_5 is popped from S as $\theta_8 < \theta_{top}$ and $x_8 < x_{top}$ and v_8 is pushedresulting S = $v_1 v_2 v_3 v_4 v_8$. Next, v_8 is popped giving S=$v_1 v_2 v_3 v_4$ as Case 4occurs for vertex v_9.

Condition 2b: If $\theta_i \ge \theta_{top}$, then v_i is pushed to S based on a flag value asfollow. Initially flag = 0. For $y_i = y_{top}$, if $\theta_{top} < 90^0$ and $\theta_i > 90^0$ (for v_0 is 90^0 vertex), then flag =1 and v_i is the offset point. Similarly, for v_0 being 270^0vertex, the condition is $y_i = y_{top}, \theta_{top} < 270^0$ and $\theta_i > 270^0$. **Case 1:** When $\theta_i \ge \theta_{top}$and flag = 0, then v_i is pushed to S.

Case 2: When $0^0 \le \theta_i \le 90^0$ andflag = 1, if $y_i < y_{offset}$, then v_i is pushed to S and flag = 0. If $v_{i-1}v_i \parallel v_0 v_1$ and $\theta_i < \theta_{top}$ and $y_{top} > y_i$ then v_{top} is popped. Whereas if $not(v_{i-1}v_i \parallel v_0 v_1)$, $\theta_i < \theta_{top}$and $x_{top} > x_i$ then v_{top} is popped. The popping of the top of S is continued till this condition is satisfied.

In Fig. 7, initially flag = 0. So, v_1, v_2 v_3 v_4 are pushed to S. At v_5 set flag to1and v_5 is offset point.At v_7, $v_{i-1}v_i \parallel v_0 v_1$ and $\theta_i < \theta_{top}$ but $y_{top} < y_i$ so v_7 is not pushed to S as flagMin = 1(case2). Similarly, v_8 is invisible and hence not pushed in S. At v_{11},again flag is set to 0 and v_{11} is pushed to S.

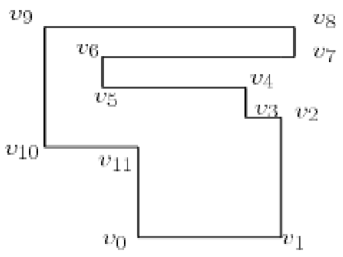


Fig7: Condition 2b

Condition 3:When $\theta_i = 0^0$ (for any vertex): A flag (flagMin) is used and initialized to 0. While $\theta_i = 0^0$ and flagMin=0, then v_i is visible. If $\alpha_i \alpha_{i+1}$ is of the pattern 12, then flagMin = 1 and v_i is the offset point. Also offsetVal = x_i (if y_i = 0) or y_i (if x_i = 0). Again, if $\theta_i = 0^0$ and flagMin = 1, following two cases may occur.Case1: When offsetVal < (x_i or y_i whichever is non-zero), v_i is not visible.Case2: When offsetVal > (x_i or y_i whichever is non-zero), v_i is visible andflagMin is set to 0. As mentioned earlier, v_i is not visible when $\theta_i = 0^0$ and α_i = 5. Hence, for $\theta_i = 0^0$ and α_i = 5, pop v_{i-1} if it is in S. Also, v_{top} is poppedif $x_{top} > x_i$. Popping is continued until $x_{top} \le x_i$.

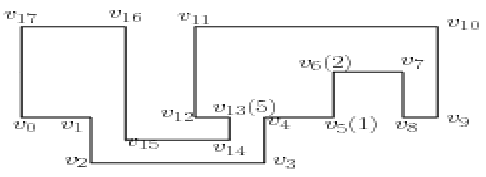
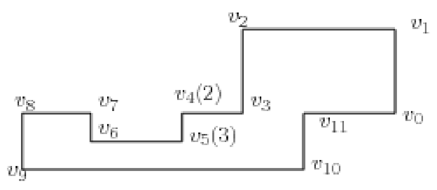


Fig8: Condition 3 (for 90^{0}vertex)

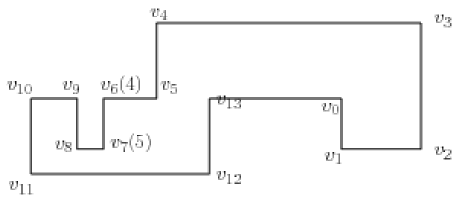
In Fig. 8, initially flagMin = 0. So, v_1 , v_4 , v_5 is pushed to S. At v_5 thepattern 12 occurs for $\alpha_5\alpha_6$. Hence, set flagMin = 1 and v_5 is offset point. v_8 is not pushed to S as flagMin = 1. Similarly, v_9 is invisible and hence not pushedin S. At v_{12} , $x_{12} < x_5$. Hence, set flagMin = 0 and push v_{12} to S. v_{13} is notpushed as $\alpha_{13} = 5$ for $\theta_{13} = 0^0$.

Condition 4:When $\theta_i = 90^0$ for v_0 being 90^{0}vertex:the procedure is similar as that of condition 3 processing except the fact that $\theta_i = 90^0$ .Also, for $\alpha_i = 5$, v_i may be visible for flagMax = 0.Here when the pattern of $\alpha_i\alpha_{i+1}$ is 23 set flagMax = 1 as shown in Fig. 9(a). In Fig.9(a), v_3, v_4 is pushed to S as flagMax= 0. At v_4 , set flagMax = 1as the pattern 23 occurs for $\alpha_4\alpha_5$ and y_4 is the offset. v_7, v_8 are not visible asflagMax = 1 at that point of traversal. As $y_{11} < y_4$, set flagMax = 0 andpush v_1 to S.

Condition 5: When $\theta_i = 270^0$ for v_0 being 270^{0}vertex: it is similar to the case of condition 4 except the fact that $\theta_i = 270^0$. Here when the pattern of $\alpha_i\alpha_{i+1}$ is 45 set flagMax = 1 (Fig. 8(b)). Initially, flagMax = 0. So push v_5, v_6 readily to S. Due to $\alpha_6\alpha_7$ having pattern 45, set flagMax = 1 and offset = y_6 . v_9, v_{10} not visible due to flagMax = 1. At v_{13}, $y_{13} < y_6$. Hence set flagMax = 0. Push v_{13} to S.



(a)



(b)

Fig9: (a) Condition 4, (b) Condition 5

3.2 Demonstration

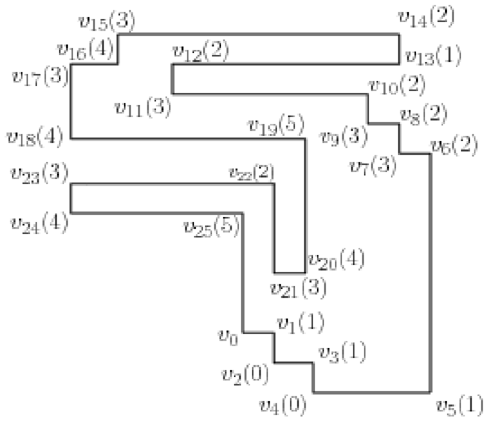


Fig10: Demonstration with an example

In Fig. 10, stating from v_0 with flag = 0, v_1 is initially pushed to stack givingS = v_1 . As $\theta_2 < 0^0$, v_2 is not visible from v_0 , hence not pushed to S(Condition 1). Similarly,vertices v_k (k = 3; 4; 5) are discarded. At v_6 , Condition 2(b) occurs with flag=0 initially.So v_6 is pushed giving S = v_1v_6 . Similarly, v_k (k = 7; 8; 9; 10) are pushed giving S = $v_1v_6v_7v_8v_9v_{10}$. At v_{11} ,flag is set to 1 as (Condition 2b) occurs with $y_{offset} = y_{11}$. While flag = 1, v_i is not pushed unless Case 2 of (Condition 2b) occurs. Hence vertices from v_{12} to v_{18} are discarded as either $\theta_i > 90^0$ or the above mentioned conditionis violated. At v_{19} , set the flag to 0 and push it to S = $v_1v_6v_7v_8v_9v_{10}v_{19}$ as $\theta_{19} > 90^0$ and $y_{offset} > y_{19}$ (Case 2 of (Condition 2b)). When traversal reaches v_{20}, v_{19} is popped. Next $v_{10}, v_9, v_8, v_7, v_6$ are popped (Condition 2a ,Case 2(a) of A. v_0 is 90^{0}vertex occurs) and v_{20} is pushed resulting S = v_1v_{20} . v_{21}, v_{22} are pushedas $\theta_i > \theta_{top}$ and flag = 0 (Case 1 of (Condition 2b)) giving S = $v_1v_{20}v_{21}v_{22}$ .Next, v_{23} and v_{24} are discarded as (Condition 1) occurs. Finally, v_{25} is pushedto S = $v_1v_{20}v_{21}v_{22}v_{25}$.

3.3 Analysis on time-complexity

For each vertex, the visibilities of other vertices are determined by traversing the polygon in an anticlockwise manner. A vertex can be pushed to the stack only once, and once a vertex is popped it is never pushed to stack again. Hence, the time taken to find the visible vertices from a given vertex is $o(n)$. So, the total time complexity for computing the visibility matrix, i.e., each vertex in turn is considered as thereference vertex and the visible vertices are found, is $o(n^2)$.

4 EXPERIMENTAL RESULTS

The proposed algorithm is implemented in C and is tested on several datasetscontaining various digital images of different shapes and forms. Fig 11(a) showsthe input image(col (i)), orthogonal polygon for the image with grid size=14(col (ii)) and thecorresponding visibility matrix(col(iii)). In Fig 11(b) to 11(h)shows the input image(col (i)&(iv)), orthogonal polygon for the image(col (ii)&(v)), and the output visibility matrices in form of image (col (iii)&(vi)) by mapping each 1 of the visibility matrix with whitepixel and each 0 with black pixel. The images are of size nxn for an orthogonalpolygon with n vertices. The black portion indicates the corresponding verticesare not visible from each other. The white areas, indicate the visibility of vertices from each other. Clique are the portions where we get a continuous white area. The output has significant importance in shape analysis and pattern matchingof digital objects. The algorithm is successfully tested on digital objects of various types and shapes. It is to be noted that the visibility matrices aresymmetric.Hence the col(iii)& (vi) outputs are all symmetric along the diagonal. The diagonal isalways white with minimum thickness of 3 i.e. the immediate vertices from anyvertex is always visible. Hence we may analyze only the triangular half of theoutput pattern images cut along the continuous white diagonal. The small white dots (alternate white and black dot(s)) represent we are going to traverse a staircase like structure where each 270^0 vertexmay be visible from one of the 270^0 reference vertex on the staircase. We may further analyze in depth the different patterns in the output and have some accurate prediction of the shape of the digital object.

5 CONCLUSION

The presented algorithm computes the visibility matrix of a given orthogonal polygon in $o(n^2)$ time complexity. The algorithm runs fast as it involves no complex computations, in fact only comparisons are required. The algorithm has a vast application area where if used, can solve many complex problems based on the problem addressed here. The algorithm is very fast, efficient and having optimal time complexity. The main contribution of the result of the algorithm is to identify the pattern, analysis of shape of digital object. This allows us to handle huge polygons with only a polynomial amount of time. The presented algorithm does not consider the inner polygons with holes; however that remains our future focus. A number of experiment results are given whichshows it works for different complex shapes of polygon. In future we may usethe result in various shape matching and pattern analysis applications.

REFERENCES

[1]Freeman H, Loutrel PP (1967) *An algorithm for the two-dimensional hidden line* problem. IEEE Trans Electronic Computers EC- 16 (6): 784-790

[2] Davis L, Benedikt M (1979) *Computational models of space: Isovists and isovist fields.* Comput Graph Image Proc 11:49 72

[3] Nilsson NJ (1969) A mobile automation: *An application of artificial intelligencetechniques.* Proc IJCAI-69, pp 509-520

[4]E1Gindy H (1985)*Hierarchical decomposition of polygons with applications*. PhD Thesis, McGill University

[5]Lee DT (1983) *Visibility of a simple polygon.* Computer Vision, Graphics, andImage Processing 22:207-221

[6]Lee DT, Chert IM (1985)*Display of visible edges of a set of convex polygons*. *In:Toussaint GT (ed*) Computational Geometry. North-Holland, pp 249-265

[7] Lee DT, Lin A (1984) *Computing the visibility polygon from an edge.* Tech Rep,Northwestern University

[8]Joe B, Simpson RB (1987)*Algorithms and correctness proofs for visibility polygon computations*, Tech Rep CS-87-03, University of Waterloo,Waterloo, Ontario,Canada

[9] Sack JR (1984*) Rectilinear computational geometry*. PhD Thesis, School of Computer Science, Carleton University, Ottawa, Ontario, Canada

[10] Avis D, Toussaint GT (1981)*An optimal algorithm for determining the visibility ofa polygon from an edge*. IEEE Trans Comput C-30:910-914

[11]Guibas L, Hershberger H, Leven L, Sharir M, Tarjan RE (1986*) Linear time algorithms for visibility and shortest path problems inside polygons*. Proc 2nd Symp Computational Geometry, Yorktown Heights, New York, pp 1-13

[12]Jeong-In Doh and Kyung-Yong Chwa *Visibility problems for orthogonal objects in two-or three-dimensions*, The Visual Computer (1988) 4:84-97

[13] M. Breen.*Clear visibility and unions of two starshaped sets in the plane*. Pacific Journal of Mathematics, 115:267,275, 1984.

[14]J. Culberson and R. Reckhow.*Orthogonally convex coverings of orthogonal polygons without holes.* Journal of Computer and Systems Science, 39:166,204, 1989.

[15]R. Motwani, A. Raghunathan, and H. Saran.*Covering orthogonal polygons with star polygons: the perfect graph approach.* Journal of Computer and Systems Science, 40:19,48, 1990.

[16]S. Schuierer and D.Wood.*Staircase visibility and computation of kernels.* Algorithmica, 14:1,26, 1995.

[17]L. Gewali, M. Keil, and S. Ntafos.*On covering orthogonal polygons with star-shaped polygons.* Information Sciences, 65:45,63, 1992.

[18]M. H. Overmars and D. Wood.*On rectangular visibility.Journal of Algorithms*,9:372, 390, 1988.

[19] P. K. Agarwal and M. Sharir.*Circular visibility from a point in a simple polygon.*International Journal of Computational Geometry and Applications, 3:1 25,1993.

[20]S.-Y. Chou and R. C. Woo.*A linear-time algorithm for constructing a circular visibility diagram.*Algorithmica, 14:203 228, 1995.

[21]J. Dean, A. Lingas, and J.-R. Sack.*Recognizing polygons, or how to spy.*The Visual Computer, 3:344 355, 1988.

[22]*Visibility algorithms in the plane by Ghosh S.K.*(CUP, 2007)(ISBN0521875749)(334s)

[23]A. Rosenfield and A. C. Kak. *Digital picture processing,* 2nd ed. Academic Press, New York, 1982.

[24]P. Bhowmick, A. Biswas, and B. B. Bhattacharya.*Isothetic polygons of a 2D object on generalized grid.* In Proc. 1st Intl. Conf. Pattern Recognition and Machine Intelligence (PReMI), LNCS, 3776:407412, 2005.

[25]A. Biswas, P. Bhowmick, and B. B. Bhattacharya. *TIPS: On finding a Tight Isothetic Polygon Shape covering a 2D object.*In Proc. 14th Scandinavian Conf. Image Analysis (SCIA), LNCS, 3540:930939, 2005

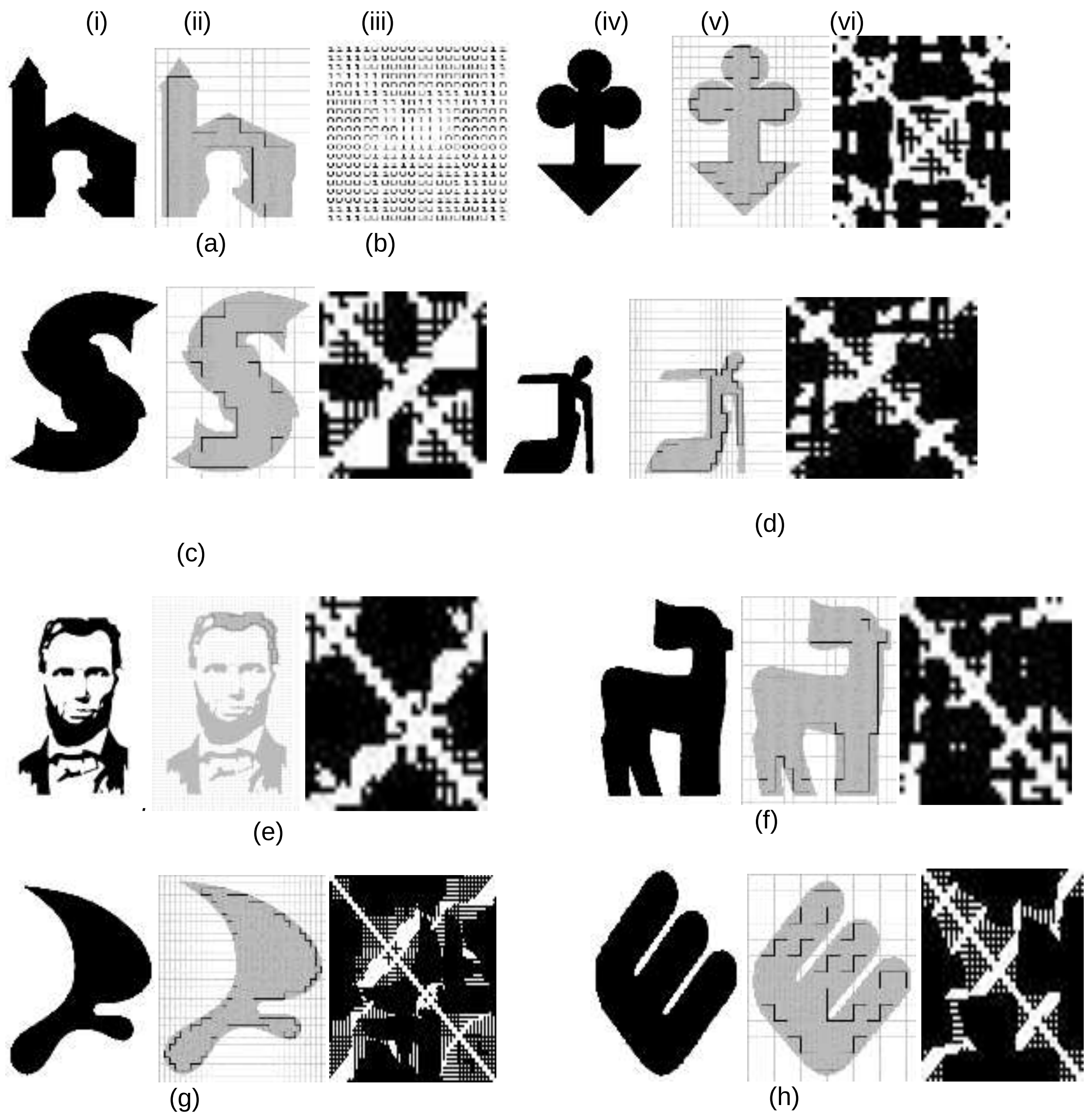


Fig 11: A set of results on different digital objects(a) showing visibility matrix, (b-h) pattern output