The Visibility Graph of Congruent Discs is Hamiltonian

David Rappaport*

School of Computing

Queen's University, Kingston, Ontario, K7L 3N6, CANADA.

Abstract

We show that the visibility graph of a set of disjoint congruent discs in $\mathbb{R}^2$ is Hamiltonian, as long as the discs are not all supported by the same line. The proof is constructive, and leads to efficient algorithms for obtaining a Hamilton circuit.

Keywords: computational geometry, Hamiltonian circuit, visibility graph

1 Introduction

Let S denote a set of non-intersecting closed convex objects in $\mathbb{R}^2$. Consider two distinct objects in S, a and b. We use $S - \{a, b\}$ to denote $\{x : x \in S \text{ and } x \notin \{a, b\}\}$. We say that the objects a and b *see each other* if there exists a straight line segment l with one point in a and one point in b such that l lies in the complement of S - $\{a, b\}$. We call such a line segment a *line of sight*. The *visibility graph* of S, denoted by Vis(S), associates a vertex to each object of S, and an edge between two vertices if and only if the associated objects see each other.

The combinatorial structure of the visibility graph for sets in $\mathbb{R}^2$ has been studied extensively. See the article by O'Rourke [10] for a comprehensive overview. Several papers present results on the combinatorial structure of the visibility graph of line segments [12, 3, 9, 8]. This paper builds upon the result of Hosono et al. [7] that the visibility graph of a set

*Accepted for publication in Computational Geometry Theory and Applications, August 13, 2002.

of translates of the same convex polygon in $\mathbb{R}^2$ contains a Hamiltonian path. We combine their algorithms with a simple pre-processing step to produce a Hamiltonian subgraph of the visibility graph of a set of disjoint congruent discs.

A graph has *Hamiltonian path* if there exists a traversal that visits every vertex exactly once. A graph has a *Hamiltonian circuit* if it has a Hamiltonian path that can be augmented with one additional edge joining the beginning to the end of the path. It is common to say that a graph is *Hamiltonian* if it has a Hamiltonian circuit. Although most often associated with the Irish Mathematician Sir William Rowan Hamilton, the interest of graphs with Hamilton's property pre-date Hamilton by centuries. Hoffman and Wolfe [5] have written an account of the history of this problem.

Hosono et al. [7] show that the visibility graph of a set of translates of the same convex polygons in $\mathbb{R}^2$ contains a Hamiltonian path. Using the algorithms of Hosono et al. [7] coupled with a simple pre-processing step we are able to obtain a Hamiltonian subgraph of the visibility graph of a set of disjoint congruent discs.

2 Preliminaries

This section will provide some of the basic definitions and notation that we use.

We use $G = (V, E)$ to denote a graph with vertex set V and edge set E with no loops or multiple edges. We denote edges by two element subsets of the vertex set. A graph $G = (V, E)$ is *planar* if the vertices V can be positioned on a plane, so that all edges E can be realized by non-crossing straight line segments. We call this a *drawing* of G. We use *plane graph* to denote a planar graph along with a drawing of it. A plane graph partitions the plane into disjoint regions, or *faces*. There is exactly one face that is unbounded, and this is called the *outer* face. A two-connected plane graph is a *triangulation* if all of its faces, except possibly the outer face, are triangles. A triangulation with an outer face that is also a triangle is called *maximal planar*. We say that $\{u, v, w\} \subseteq V$ is a *3-clique* if $\{\{u, v\}, \{v, w\}, \{u, w\}\} \subseteq E$. A 3-clique in a triangulation is called a *separating triangle* if it contains vertices both in its interior and exterior. We call a triangulation an *NST triangulation* when it contains no separating triangles. Given a plane graph, G, the *boundary graph*, B, is a subgraph of G induced by the vertices of the outer face. A *boundary-*

chord in a boundary graph is any edge of the boundary graph which is not an edge of the outer face of G. Figure 1 depicts an NST triangulation and its associated boundary graph.

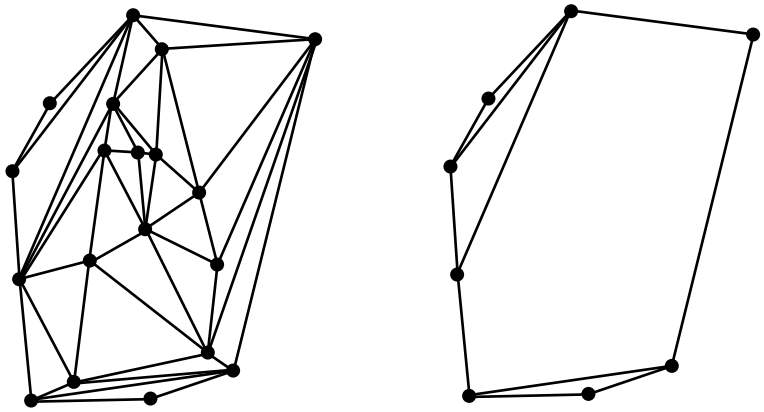

Figure 1: An NST triangulation and its associated boundary graph.

In a paper published in 1931, Hassler Whitney [15] proved that maximal planar graphs with no separating triangles are Hamiltonian. The proof uses a recursive algorithm to construct a Hamiltonian path between two adjacent vertices, thus leading to a Hamiltonian circuit. The algorithm makes use of the following lemma

Lemma 2.1 (Whitney's lemma) *If the boundary graph of an NST triangulation has vertices A, B, C, that defines the following labelling $A = a_0, a_1, \ldots, a_\alpha = B = b_0, b_1, \ldots, b_\beta = C = c_0, c_1, c_2, \ldots c_\gamma = A$, so that there are no boundary-chords of the form $\{a_i, a_j\}$, $\{b_i, b_j\}$, or $\{c_i, c_j\}$, then there is a Hamiltonian path from A to B.*

In 1984 a new simpler proof of Whitney's lemma is used to design an algorithm that finds a Hamiltonian cycle in an NST triangulation in linear time [1]. Dillencourt [4] subsequently extended Whitney's results in several ways. We make use of the following lemma.

Lemma 2.2 (Dillencourt's lemma) *If the boundary graph of an NST triangulation has no face using more than three boundary chords then the graph is Hamiltonian.*

In this paper we show that the visibility graph of congruent discs contains a planar subgraph that satisfies Dillencourt's lemma. The main result of this paper is given by the following Theorem.

Theorem 2.3 *The visibility graph of a set of congruent discs in the plane, not all supported by the same line, is Hamiltonian.*

3 Methods

The results obtained in this paper adds a pre-processing step to the algorithm described by Hosono et al. [7]. A sketch of the algorithm of Hosono et al. [7] to find a subgraph of the visibility graph of a set of suitably oriented translates of a convex body is given , followed by some needed details.

Let M represent a graph such that the vertex labels in M are in a one-to-one correspondence with a set S of congruent discs. Assume that the input set of discs does not have all discs supported by the same line. Furthermore, without loss of generality, assume that no two discs are supported by the same horizontal or vertical line. The edges of M will fall into three classes. M contains a class I edge $\{v, w\}$ whenever there is a horizontal line of sight between the discs v and w. M contains a class II edge $\{v, w\}$ whenever there is a horizontal line H that partitions S into two disjoint subsets one entirely above and the other entirely below H, so that v is the lowest disc in the upper subset and w is the highest disc in the lower subset. The remaining edges of M, those of class III, will be added by an iterative process. Before describing this process we need to introduce some geometric terminology.

Let $l : ax + by = c$ denote an infinite line. Let l^+ denote the closed half-plane bounded by l and satisfying $ax + by \geq c$. Let l^-, denote the open half-plane satisfying $ax + by < c$. Let Γ denote an arbitrary closed subset of $\mathbb{R}^2$. We say that l *supports* Γ if $l \cap \Gamma \neq \emptyset$ and, $l^+ \cap \Gamma = \Gamma$. Note, we always use $l^+(\Gamma)$ to denote the closed half plane that contains Γ and $l^-(\Gamma)$ to denote the open half plane that is disjoint from Γ. If $\gamma \in \Gamma$ is contained in l then we say that γ is *extreme*.

For a disc s we define $\textsc{North}(s)$ as a horizontal support line above s, and north(s) as the extreme point of s contained in $\textsc{North}(s)$. Similarly for S a set of distinct congruent discs $\textsc{North}(S)$ is a horizontal support line for the union of the discs of S, and north(S) is the leftmost extreme disc contained in $\textsc{North}(S)$.

By analogy we define support lines, and extreme elements of a set Γ with respect to the other compass points, using $\textsc{South}(\Gamma)$, south(Γ), $\textsc{East}(\Gamma)$, east(Γ), $\textsc{West}(\Gamma)$, and

west(Γ).

For an example of how this terminology is used consider the class II edge {v,w} as above, and observe that the area bounded by $\textsc{South}^-(v)$ and $\textsc{North}^+(w)$ is in the complement of S. This area can be descriptively denoted as a *clear corridor*. Furthermore, in Figure 2 north(S), south(S), east(S), and west(S) are respectively labelled N, S, E, W.

We say that s is maximal in the east direction, or *east-most*, if there exists a horizontal line l such that $l \cap s$ contains the rightmost point in a non-empty intersection of l and S. Let r, s, and t denote three distinct east-most elements of S with the following properties.

- north(s) is above north(r) and north(r) is above north(t)
- west(r) is to the left of both west(s) and west(t)
- $\{s, r\}$ and $\{r, t\}$ are edges {class I, II, or III } in M

These conditions characterize class III edges $\{s, t\}$ that are added to M. A consequence of this characterization is that the region bounded by $\textsc{East}^-(r) \cap \textsc{South}^-(s) \cap \textsc{North}^-(t)$ is non-empty and is contained in the complement of S. This property has been proved by Hosono et al. [7]. We describe this region as a *clear half-corridor*. In a similar symmetric way class III edges may also be characterized by west-most triples. For now we depict the structure of this type of graph in Figure 2. The precise algorithmic details follow.

The algorithm uses the sweeping line technique common to many algorithms in computational geometry [2, 11]. The algorithm sweeps a horizontal line from the top to the bottom of the plane. A priority queue, Q, using points from $\{\text{north}(s), \text{south}(s) | s \in S\}$ sorted by decreasing y-coordinate determines the discrete events that signal some action. Note: Whenever north(t) and south(s) have the same y-coordinate we queue north(t) before south(s). We assume that the input set of discs does not have all discs supported by the same line. Furthermore, without loss of generality we assume that no two discs are supported by the same horizontal or vertical line.

A search structure L maintains in x-coordinate order all objects currently stabbed by the sweep line. We initialize L with dummy leftmost and rightmost entries, which we call α and ω. The algorithm also makes use of a pair of initially empty stacks called EastStack and WestStack that keep track of east-most and west-most objects used in the construction of class III edges. A procedure Make-III is used to handle the creation of class III edges.

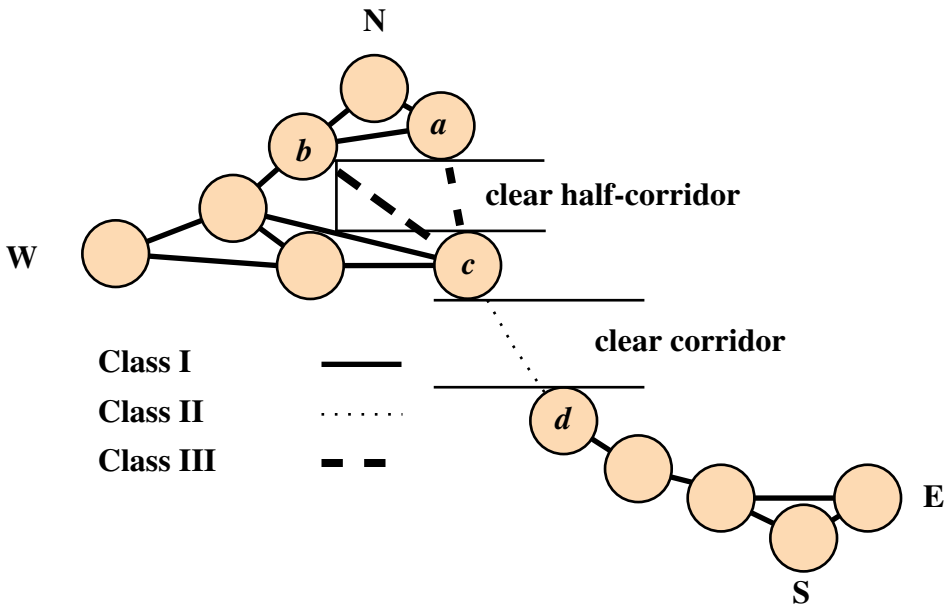

Figure 2: A sample set of discs with the edges of graph M as constructed by the algorithm Make-M. An empty half-corridor $\textsc{East}^-(b) \cap \textsc{South}^-(a) \cap \textsc{North}^-(c)$, and an empty corridor $\textsc{South}^-(c) \cap \textsc{North}^-(d)$ are shown.

Make-M(Q,L)

1. **loop**
2. $q \leftarrow$ D*equeue*(Q)
3. **case** q **of**
4. $q =$ north(s):
5. Insert s into L

   {Let the left and right neighbours of s in L be respectively denoted as s^- and s^+.}
6. **if (** $s^- = \alpha$ **) then** Make-III(WestStack,s) **else** Add class I edge $\{s, s^-\}$ to M
7. **if (** $s^+ = \omega$ **) then** Make-III(EastStack,s) **else** Add class I edge $\{s, s^+\}$ to M
8. $q =$ south(s):
9. Delete s from L.
10. **if (** $s =$ south(S) **) then return**

    {Let the former left and right neighbours of s in L be respectively denoted as s^- and s^+.}
11. **if (** $s^- = \alpha$ **and** $s^+ = \omega$ **) then** Add the class II edge $\{s, u\}$ to M, where north(u) is the next object in Q.
12. **if (** $s^- = \alpha$ **and** $s^+ \neq \omega$ **) then** Make-III(WestStack,s^+)
13. **if (** $s^- \neq \alpha$ **and** $s^+ = \omega$ **) then** Make-III(EastStack,s^-)
14. **if (** $s^- \neq \alpha$ **and** $s^+ \neq \omega$ **) then** Add class I edge $\{s^-, s^+\}$ to M

end loop

We now give the pseudo code for procedure Make-III used to create the class III edges. We use standard stack operation Push and Pop. We augment the standard stack operations with an operation Size(Stack) that returns the number of elements on the stack. We use the notation Stack(top) to obtain the value of the top element of the stack, without popping it from the stack. Similarly Stack(top-1) is used to access the value of the next to top element of the stack. The procedure Make-III loops as long as class III edges can be added to M.

```
Make-III (Stack,t)
1. loop
2.   if ( Size(Stack) < 2 ) then  Push (Stack,t) return
3.   r ← Stack(top) s ← Stack(top-1)
4.   if ( Stack = EastStack ) then  l_r = EAST(r) else l_r = WEST(r)
5.   if (  l_r^- ∩ s ≠ ∅ and l_r^- ∩ t ≠ ∅ ) then
6.      Add the edge {s,t} to M
7.      Pop(Stack)
     else
8.      Push(Stack,t) return
   end loop
```

Observe that our algorithm is *iso-dependent*, that is, the output varies according to the underlying orientation of the set of discs. An example that makes this point obvious can be seen in Figure 3.

Graph M is a subgraph of Vis(S) [7] and Figure 2 depicts its structure. A planar drawing of M is obtained by positioning vertices at the centres of the discs. Every face in this drawing of M is a triangle except for the outer face, and no triangle is a separating triangle and as was proved by Hosono et al. [7] no triangle is a separating triangle. We can describe the outer face of M by four paths. Beginning at east(S) and proceeding in a counter clockwise direction we have, $P_1 = [\text{east(S)}, \ldots, \text{north(S)}]$, $P_2 = [\text{north(S)}, \ldots, \text{west(S)}]$, $P_3 = [\text{west(S)}, \ldots, \text{south(S)}]$, $P_4 = [\text{south(S)}, \ldots, \text{east(S)}]$. Observe that these paths are overlapping, and all contain at least one vertex. Since the outer face may not be two-connected some vertices other than the compass points may appear on more than one path. Given a disc $s \in S$, we say that s is *maximal with respect to the first quadrant* if the region $\text{NORTH}^-(s) \cap \text{EAST}^-(s)$ is in the complement of S. We can similarly define maximality with respect to the other quadrants.

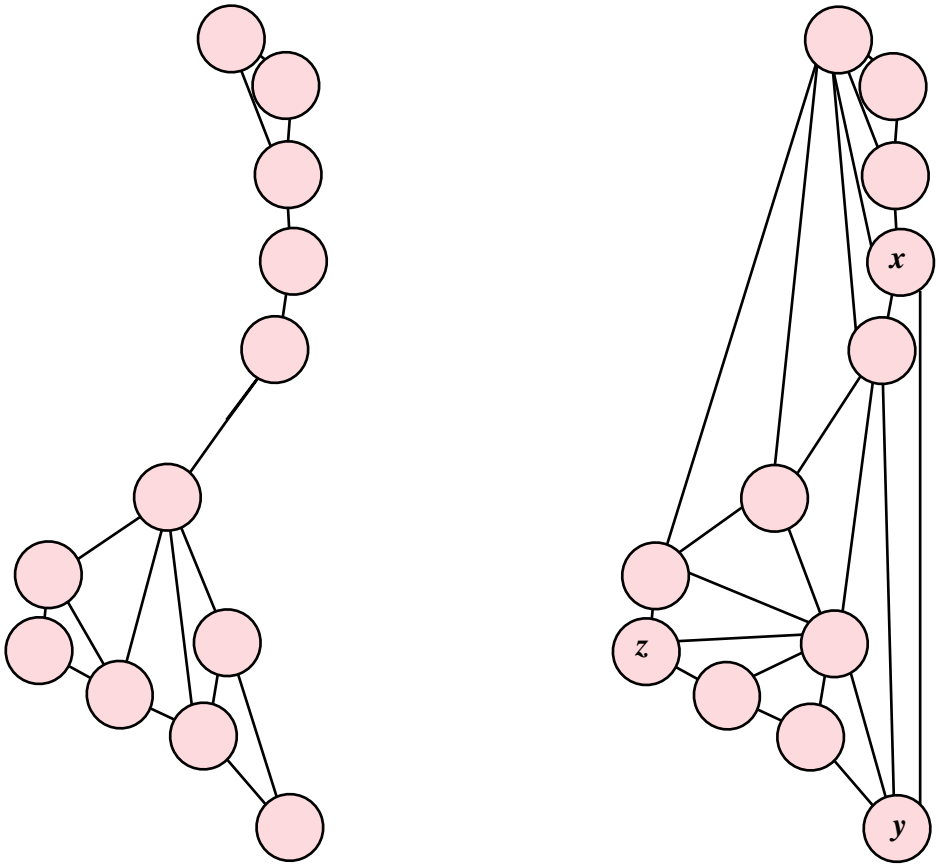

Figure 3: A rotated version of the set S from Figure 2, with the previously computed edges of M (left) and the newly computed edges of M (right). Later discussion will focus on discs x, y, and z.

Lemma 3.1 (see [7]) *All objects on P_i are maximal with respect to quadrant i.*

Lemma 3.1 implies that the graph M has no intra-path chords: all boundary-chords of the boundary graph of M connect distinct vertices on different paths. We call the union of the paths P_1 and P_4 the *eastern frontier* and the union of the paths P_2 and P_3 the *western frontier*.

The graph M as currently constructed does not necessarily satisfy the conditions of our theorem: that the graph be two-connected, that it satisfy Dillencourt's lemma, and that the boundary of the graph be decomposable into at most three paths with no intra-path chords. So that graph M will meet these goals, we first compute a favourable orientation from which to apply algorithm Make-M.

The proper orientation is found by using the convex hull of the discs [13]. Since the discs are congruent, we can simply find the convex hull of the disc centres, using any convex hull algorithm for planar point sites [2, 11]. Let $c(x)$ denote the centre of disc x and $CH(S)$ denote the convex hull of the disc centres of S. Identify three of elements$c(x), c(y), c(z)$ of $CH(S)$ with the following properties:

- The elements $c(x), c(y)$ are adjacent on the hull, that is the line segment $c(x)c(y)$ is a convex hull edge.
- There exists lines $l1$ and $l2$ disjoint parallel lines of support of $CH(S)$ such that $l1$ passes through the convex hull edge $c(x)c(y)$ and the line $l2$ passes through $c(z)$.
- There is a line $l3$ perpendicular to $l1$ that passes through $c(z)$ and intersects the segment $c(x)c(y)$.

The existence of such points for any convex polygon of four or more points, has been shown in the context of clamping a polygon [14] and of determining the width of a polygon [6]. We now orient our coordinate system so that $l1$ and $l2$ are vertical, and $l3$ is horizontal. We also translate $l1$ and $l2$ so that they become supporting lines of S. Without loss of generality we say $l1$ is $\textsc{East}(S)$, with x above y and $l2$ is $\textsc{West}(S)$. We may now rotate our coordinate system by a minuscule amount so that x is uniquely east(S) and remains above y but no two discs are supported by the same horizontal or vertical line. This minuscule rotation can be determined by examining all pairs of discs. In terms of efficient computation, we need not rotate, rather we only need to simulate the rotation in a consistent way when discs are compared in algorithm Make-M.

We prove that after re-orienting the set of discs and applying algorithm Make-M we obtain a graph with the requisite properties.

Lemma 3.2 *After re-orienting the discs the graph M is two-connected.*

Proof: Recall that we obtain a planar drawing of M by placing vertices at the centres of the discs, so that every face in this drawing of M is a triangle except for possibly the outer face. Thus if M is not two-connected then it is the boundary of the outer face that is not two-connected. Let $\delta(M)$ denote the subgraph of M containing just the vertices and

edges that belong to the outer face of M. Let us assume that vertex v is a cut point of $\delta(M)$. Thus a traversal of the cycle $\delta(M)$ must pass through v at least twice. It follows from lemma 3.1 that v must be maximal both on the eastern and western frontiers, and v must be maximal with respect to two quadrants. That is, v is either maximal with respect to quadrants one and three, or quadrants two and four. We assume that v is maximal with respect to quadrants one and three, noting that the other case is symmetric. We show that this assumption leads to a contradiction. The centre of v, $c(v)$ is within the region $l1^+$ and $l2^+$. If v is above the line $l3$ then $c(z)$ is in the third quadrant of $c(v)$, so v cannot be maximal with respect to the third quadrant. If $c(v)$ is below the line $l3$ then $c(x)$ is in the first quadrant of $c(v)$, so v cannot be maximal with respect to the first quadrant. This contradicts our assumption that v is a cut point. Thus we conclude that the graph M is two-connected.

□

Observe that a crucial element of the preceding proof was the location of x, y and z. This delineates the necessity of obtaining a favourable orientation. Positioning x, y, and z also plays a role in proving that no face in the boundary graph of M uses more than three boundary-chords.

Lemma 3.3 *There are no boundary-chords i, j with i in P_1, and j in P_4 in the graph M.*

Proof: If i is in P_1 then i is either the disc we labeled x or a disc that precedes x in a traversal of $\delta(M)$ from north(S) to x. Similarly j is either y or a disc that succeeds y in a traversal of $\delta(M)$ from y to south(S). Thus if i, j is a boundary-chord then the following properties hold:

1. SOUTH(i) is above NORTH(j)
2. $c(z)$ is below SOUTH(i) and above NORTH(j)
3. At least one the following statements are true: $c(x)$ is below SOUTH(i) or $c(y)$ is above NORTH(j)

We enumerate through the classes of edges in M and show that any boundary-chord $\{i, j\}$ with i in P_1, and j in P_4 leads to a contradiction. The edge $\{i, j\}$ cannot be a class I edge because of property 1. The edge $\{i, j\}$ cannot be a class II edge because of property 2.

Properties 1 and 2 together with the fact that x, y and z are incident to vertical supporting lines of S ensure that there is no clear half-corridor supported by SOUTH(i) and NORTH(j). So the edge $\{i, j\}$ cannot be a class III edge. This leads to the conclusion that there are no boundary-chords with end points in P_1 and P_4. □

Lemma 3.4 *No face in the boundary graph of M has more than three boundary-chords.*

Proof: Let boundary-chords that cross from the eastern frontier to the western frontier be called *east-west boundary-chords*. Using the fact that the boundary graph is planar and two connected we can make the following statements. No face in the boundary graph can have more than two east-west boundary-chords. Also no face in the boundary graph can have more that two boundary-chords that cross from the from P_2 to P_3. We call these boundary-chords *north-south boundary-chords*. If a face has two north-south boundary-chords then it can have no east-west boundary-chords. If the face has one or more east-west boundary-chord then it can have no more than one north-south boundary-chord. Therefore no face in the boundary graph of M has more than three boundary-chords. □

Now we can apply Dillencourt's lemma to obtain theorem 2.3.

4 Discussion

We have shown that the visibility graph of a set of disjoint congruent discs is Hamiltonian. Hosono et al. [7] show that visibility graph of a set of translates of a convex polygon admits a Hamilton Path. However, the proof falls short when it comes to Hamilton Circuits. The results differ because for convex translates we are required to use an orientation based on the shape of the convex polygon. For a set of congruent discs we are free to choose a favourable orientation.

Our proof is constructive, and does not explicitly use the visibility graph. Consider a set S of n disjoint congruent discs. The graph that we construct has complexity in O(n), whereas the visibility graph of S may be a complete graph , that is, it has O(n^2) edges. { Observe that Vis(S) is a complete graph when the disc centres are at the vertices of a sufficiently large convex polygon. }. It is straightforward to show that the worst case time computational complexity of algorithm Make-M is in $O(n \log n)$. Furthermore, Dillencourt [4] has established that the linear algorithm of Asano et al. [1] applies in this situation too.

Thus, given a set of n disjoint and congruent discs, a Hamiltonian circuit using visibility graph edges can be found in $O(n \log n)$ time.

Acknowledgments

I am indebted to the referees whose comments have helped me improve the presentation of these results. This research was supported by NSERC of Canada.

References

[1] Ta. Asano, S. Kikuchi, N. Saito, *A linear algorithm for finding Hamiltonian cycles in 4-connected maximal planar graphs,* Discrete Appl. Math., 7 (1984) 1-15

[2] Mark de Berg, Marc van Kreveld, Mark Overmars, Otfried Schwarzkopf, Computational Geometry: Algorithms and Applications, Springer-Verlag (1997).

[3] P. Bose, G. Toussaint, *Growing a tree from its branches,* Proc. First Pacific Conf. Computer Graphics Appl., Seoul, Korea , vol. 1 , World Scientific Publishing Co. (1993) 90–99.

[4] M. B. Dillencourt, *Hamiltonian cycles in planar triangulations with no separating triangles,* J. Graph Theory, 14 (1990) 31–49.

[5] A.J. Hoffman and P. Wolfe, *History* The Traveling Salesman Problem, Wiley, (1985).

[6] M. E. Houle and G. T. Toussaint, *Computing the width of a set,* IEEE Trans. Pattern Anal. Mach. Intell. PAMI-10, 5 (1988) 761–765.

[7] K. Hosono, H.Meijer, and D. Rappaport, *On the visibility graph of convex translates*, Discrete and Applied Mathematics, 113 (2-3) (2001) 195–210.

[8] K. Hosono and K. Matsuda, *On the perfect matching of disjoint compact sets by noncrossing line segments in* $\mathbb{R}^n$, Discrete Applied Mathematics 118 (2002) 223–238.

[9] A. Mirzaian, *Hamiltonian triangulations and circumscribing polygons of disjoint line segments,* Comput. Geom. Theory Appl., 2, 1 (1992) 15–30.

[10] J. O'Rourke, *Visibility*, Handbook of Discrete and Computational Geometry, CRC Press (1997) 467–479.

[11] J. O'Rourke, Computational Geometry in C, 2nd Ed. Cambridge University Press (1998).

[12] D. Rappaport *Computing simple circuits from a set of line segments is NP-complete,* SIAM J. Comput., 18, 6, (1989) 1128–1139.

[13] D. Rappaport, *A convex hull algorithm for discs, and applications*, Comput. Geom. Theory Appl., 1, 3 (1992) 171–181.

[14] D. L. Souvaine and C. J. Van Wyk *Clamping a polygon,* Visual Comput., 10 (1994) 484–494.

[15] H. Whitney, *A theorem on graphs,* Ann. Math. vol. 32, (1931) 378-390.