On Hamiltonian Triangulations in Simple Polygons

(Extended Abstract)

Giri NARASIMHAN *

Department of Mathematical Sciences
The University of Memphis
Memphis TN 38152.
e-mail: giri@next1.msci.memphis.edu

Abstract. An n-vertex simple polygon P is said to have a *Hamiltonian Triangulation* if it has a triangulation whose dual graph contains a Hamiltonian path. Such triangulations are useful in fast rendering engines in Computer Graphics. We give a new characterization of polygons with Hamiltonian triangulations. and use it to devise $O(n \log n)$-time algorithms to recognize such polygons.

1 Introduction

A simple polygon P is said to have a *Hamiltonian Triangulation* if it has a triangulation whose dual graph contains a Hamiltonian path. Such triangulations are useful in fast rendering engines in Computer Graphics, since most visualization of surfaces is currently done via triangles [AHMS, ESV]. In order to transmit triangulation data for a simple polygon to a rendering engine, one needs to send information about each of its triangular faces. For a polygon with a hamiltonian triangulation, its triangulation data can be sent at a faster rate since consecutive triangles share a face, and hence only incremental changes need to be specified per triangle. Benefits include potential reduction in rendering time by a factor of three, and compression for storing models. *OpenGL*, a triangular-mesh rendering product from Silicon Graphics uses such a specification to render hamiltonian triangulations. In a related work, Evans et. al [ESV] report a program for generating triangular strips from polygonal models. Since interactive display rates are becoming increasingly important in virtual reality and visualization applications, many rendering engines use special-purpose hardware to speed up the rendering process by maintaining a buffer to store k previously transmitted vertices. For polygons with hamiltonian triangulations, the size of the buffer needed is minimum since only the last 2 transmitted vertices need to be stored.

Hence it would be useful to recognize such polygons, especially since the recognition need be done only once, while the rendering may be performed many times. The first step towards the recognition of polygons with hamiltonian triangulations is a characterization of such polygons. Arkin et al. [AHMS] observed

* Supported in part by NSF Grant CCR-940-9752

that a polygon has a hamiltonian triangulation if and only if it is *discretely straight walkable.*

They introduced the concept of *Discretely Straight Walkability*, which is a discrete version of the *Straight Walkability* concept introduced by Icking and Klein [IK]. The original *walkability* concepts grew out of research on the *two-guards problem*, which in turn arose from the study of different versions of *art-gallery problems* (for example see [O87]). Briefly, the two-guards problem involves determining whether two guards can move along the boundary of a simple polygonal room from a common starting point s to a common destination point t while staying mutually visible at all times. Straight walkability deals with a version of the two-guards problem where the guards do not backtrack during their walk. Discrete straight walkability deals with a variant of straight walkability where only one guard moves at any one time while the other guard is stationary at a vertex. Guard problems are important because they contribute to the study of visibility in polygons. It is interesting to note its relationship to hamiltonian triangulations in polygons as well.

Using the connections to discrete straight walkability, Arkin et al. showed an algorithm to recognize polygons with hamiltonian triangulations in time that is linear in the size of the visibility graph of the given polygon P. The size of the visibility graph of P could be quadratic in the size of P and hence their algorithm could be very inefficient even for nearly convex polygons.

In section 4 we show a new characterization of discretely straight walkable polygons. Using this characterization we show more efficient algorithms for many problems related to such polygons. In particular we show (in section 5) the following results:

1. An $O(n \log n)$-time algorithm to recognize polygons with a Hamiltonian triangulation.
2. An $O(n \log n)$-time algorithm to construct such a triangulation.
3. Given vertices p and q on a simple polygon, an $O(n)$-time algorithm to determine whether the polygon is discretely straight walkable with respect to the two vertices.
4. An $O(n \log n)$-time algorithm to list out all pairs of points on a simple polygon with respect to which the polygon is discretely straight walkable.

The algorithms in [AHMS] for the first two problems had a time complexity of $O(|E|)$, where E is the set of edges in a visibility graph of the polygon. The first two results improve on the best-known time complexity for the problems. The last result subsumes the first result quoted above. The reason three of the four algorithms run in $O(n \log n)$ time instead of linear time is that the algorithms compute all the ray shots obtained by extending the edges. There does not seem to be any way of avoiding the computation of possibly redundant ray shots.

As observed in [AHMS], even if a triangulation is hamiltonian, the topology of the triangulation is not necessarily specified by an encoding sequence of vertices. A variant of the concept of hamiltonian triangulations called *Sequential Triangulations* was introduced in [AHMS]. It turns out that even in *OpenGL*, sequential triangulations can be more efficiently specified than hamiltonian triangulations. We show that any hamiltonian triangulation can be converted into a sequential triangulation by the introduction of at most $n/2$ steiner points.

2 Preliminaries

We define notation for this paper; much of our notation is borrowed from [IK, H]. A *polygonal chain* in the plane is a concatenation of line segments. The endpoints of the segments are called *vertices*, and the segments themselves are *edges*. If the segments intersect only at the endpoints of adjacent segments, then the chain is *simple*, and if a polygonal chain is closed we call it a *polygon*. In this paper, we deal only with simple polygons. Two points in the polygon are (mutually) *visible* if the line joining them is contained in the polygon. We assume that the input is in general position, which means that no three vertices are collinear, and no three lines defined by edges intersect in a common point. If x and y are points of P, then $P_{CW}(x, y)$ ($P_{CCW}(x, y)$) is the subchain obtained by traversing P clockwise (counterclockwise) from x to y.

In most of the problems we consider, two vertices of P are specially designated as the start vertex, s, and the goal vertex, t. We refer to the subchains $P_{CW}(s,t)$ and $P_{CCW}(s,t)$ as L and R, respectively; both L and R are oriented from s to t. Points on L (R) are denoted by p, p', p_1, etc. (q, q', q_1, etc.). If p is a vertex of a polygonal chain, $Succ(p)$ represents the vertex of the chain immediately succeeding p, and $Pred(p)$ the vertex preceding p. For $p, p' \in L$, we say that *p precedes p'* (and *p' succeeds p*) if we encounter p before p' when traversing L from s to t. We write $p < p'$.

A *(general) walk* is a pair of continuous functions

$$f : [0,1] \to L, \quad g : [0,1] \to R,$$

where $f(0) = g(0) = s$, $f(1) = g(1) = t$, and $f(t)$ and $g(t)$ are co-visible for all t. A walk is *straight* if f and g are monotonic functions; that is, $f(t_1) \leq f(t_2)$ and $g(t_1) \leq g(t_2)$ whenever $t_1 < t_2$.

For points x and y, $d(x, y)$ is the direction of the directed segment from x to y. The (undirected) segment between x and y is $\overline{xy}$. The ray $\mathbf{r}(x, y)$ is the ray with terminus x in direction $d(x, y)$. The line containing x and y is denoted $\ell(x, y)$, and the directed line from x through y is $\boldsymbol{\ell}(x, y)$. An important definition is that of *(ray) shots*: the *backward (ray) shot* (or *hit point*) from a reflex vertex $p \in L$, denoted $Backw(p)$, is the first point of P encountered by a "bullet" shot from p in direction $d(Succ(p), p)$, and the *forward (ray) shot* $Forw(p)$ is the point encountered by the bullet shot from p in direction $d(Pred(p), p)$. We let $d(Backw(p))$ represent the direction of the backward shot from p, i.e. $d(Succ(p), p)$. We define the backward shot $Backw(q)$ and forward shot $Forw(q)$ for a reflex vertex $q \in R$ in similar fashion.

The *visibility graph* of is the graph obtained by representing each of the vertices of the polygon by vertices in the graph and having an edge between two vertices in the graph if the corresponding two vertices in the polygon are visible.

3 Previous Work: Straight Walkability

In order to explain our results, we repeat the two-guards problem studied by Icking and Klein [IK]. Given a simple polygon, P with n vertices and two distinguished vertices s and t, determine if two guards can move from s to t along the two boundary chains while remaining mutually visible at all times. If such a movement is possible then the polygon P is said to be *walkable* with respect to s and t. Furthermore, if it is possible for the two guards to reach the destination t without ever backtracking along the way, then the polygon P is said to be *straight walkable* with respect to s and t. Additionally, if it is possible for the two guards to move from s to t in such a way that when one guard is moving the other guard is stationary at a vertex, then the polygon is said to be *discretely straight walkable* with respect to s and t. We will simply use the terms walkable polygon, straight walkable polygon, or discretely straight walkable polygon to mean that there exists a pair of vertices on the polygon with respect to which it is walkable, straight walkable, or discretely straight walkable, respectively.

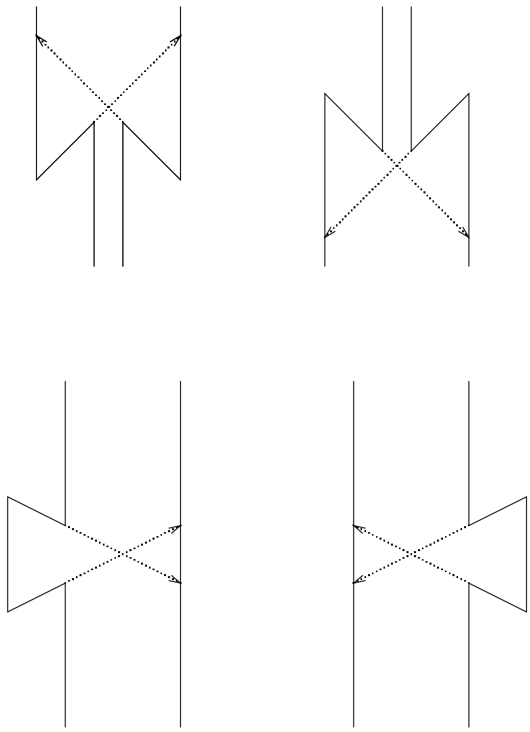
A simple polygon is said to be LR-visible with respect to two points on its boundary if the two chains determined by the two points are weakly visible from each other, i.e., every point on one chain is visible from some point on the other. As before, we will simply use the term LR-visible polygon to refer to a simple polygon that is LR-visible with respect to some two points on the polygon.

The main result of Icking and Klein's paper [IK] deals with a precise characterization of straight walkable polygons. They consider two type of forbidden configurations, known as *deadlocks* and *wedges*, which are shown in Fig. 3. Each of the cases is shown consisting of the two polygonal chains L and R and the ray shots going from one chain to the other that causes the forbidden configuration. The first two cases show a deadlock with respect to s and t respectively. The next two cases show a wedge configuration on L and R respectively. Note that in all the four configurations the two ray shots (dotted) must intersect in the interior of the polygon). It is easy to see that an instance of each of the configurations prevents a polygon from being straight walkable; [IK] show that the absence of these configurations is also sufficient to ensure straight walkability of a LR-visible polygon. This result is restated in theorem 1.

Theorem 1 *[IK] A simple polygon P is* straight walkable *with respect to two of its vertices s and t if and only if:*

1. *P is LR-visible with respect to s and t,*
2. *P does not have any* deadlocks *with respect to either s or t,*
3. *P does not have any* wedges *on the two boundary chains L and R.*

In the process of developing an algorithm to recognize straight walkable polygons, [IK] developed a considerable collection of definitions, functions and theorems, some of which are summarized here since these are necessary for our work. They define functions *lo* and *hi* on the vertices of P. Specifically, the following definition is given for L, with a symmetric definition applying to R. (The operations *min* and *max* are defined with respect to the ordering on the chain L, so that *min* of a set of points is a point of the set not preceded by any other.)



(a) Deadlocks

(b) Wedges

Fig. 1. Forbidden configurations for straight walkable polygons

Definition 1 *[IK] for a vertex $p \in L$, we define:*

$$\begin{aligned}
hiP(p) &= \min\{q \mid q \text{ vertex of } R \text{ and } L \ni Backw(q) > p\} \\
hiS(p) &= \min\{Forw(p') \in R \mid p' \text{ vertex of } L_{>p}\} \\
hi(p) &= \min\{hiP(p), hiS(p), g\} \\
loP(p) &= \max\{q \mid q \text{ vertex of } R \text{ and } L \ni Forw(q) < p\} \\
loS(p) &= \max\{Backw(p') \in R \mid p' \text{ vertex of } L_{<p}\} \\
lo(p) &= \max\{loP(p), loS(p), s\}
\end{aligned}$$

We can think of hi and lo as functions from the vertices of L to the points of R, and also from the vertices of R to the points of L. It is important to note that these are monotonic functions. [IK] are able to show that a polygon P is straight walkable if and only if $[lo(v), hi(v)]$ is a non-empty interval for every vertex v. Furthermore, if P is straight walkable, then the set of possible *walk partners* for a vertex v is precisely the points of $[lo(v), hi(v)]$, where two points $p \in L$, $q \in R$ are walk partners in a given straight walk if the two guards are at points p and q at some moment of the walk.

4 Characterization

Theorem 2 gives a precise characterization of discretely straight walkable polygons, i.e., polygons with hamiltonian triangulations. For the characterization, we consider a variant of the *wedge* configuration, which we call as the *semi-wedge* configuration. Fig. 4 shows a semi-wedge formed on the chain L. A similar one for R is not shown. Again, note that the two ray shots (dotted) must either intersect in the interior of the polygon or they must terminate in the interior of the same edge of the polygon. We show that this is a forbidden configuration for discretely straight walkable polygons. We further go on to show that this condition is also sufficient for a straight walkable polygon to be discretely straight walkable.

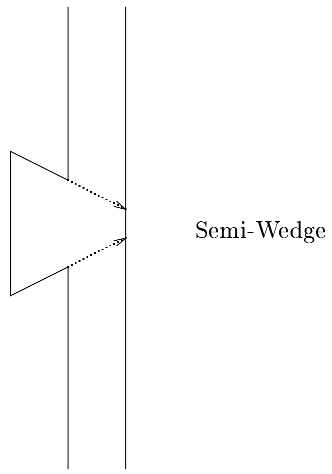


Fig. 2. Forbidden configuration for discretely straight walkable polygons

Hence a stricter version of the wedge condition is enough to ensure discrete straight walkability. An alternate characterization of discretely straight walkable polygons is the following. A polygon P is discretely straight walkable if and only if $[lo(v), hi(v)]$ contains a vertex of the polygon for each vertex v. The proof of theorem 2 also proves this alternate characterization. Compare this characterization to the lemma in [IK], which shows that the polygon P is straight walkable if and only if $[lo(v), hi(v)]$ is nonempty for each vertex v. The proof of theorem 2 also provides ideas for constructing a discrete straight walk (and consequently a hamiltonian triangulation), if one exists. This is because the vertices in the interval $[lo(v), hi(v)]$ provide a set of possible walk partners for the vertex v.

Theorem 2 *A simple polygon P is* discretely straight walkable *with respect to two of its vertices s and t if and only if:*

1. *P is straight walkable with respect to s and t,*
2. *P does not have any* semi-wedges *on the two boundary chains L and R.*

Proof. It is clear that an instance of a semi-wedge prevents a polygon from being discretely straight walkable, since a vertex inside the semi-wedge would not have a walk partner.

As a first step to prove sufficiency, assume that vertex v is such that $[lo(v), hi(v)]$ is non-empty, but devoid of vertices. This implies that $lo(v)$ and $hi(v)$ are not vertices either, and that they lie on a common edge of the polygon. Furthermore, this also implies $lo(v) = loS(v)$ and $hi(v) = hiS(v)$, since loP and hiP take only vertex values. But this is precisely a semi-wedge configuration, and hence the contradiction.

The proof will be complete if we can show that discrete straight walkability is ensured if for every vertex v, $[lo(v), hi(v)]$ has at least one vertex of P. Assume that it does not. This can only happen if for adjacent vertices v_i, v_{i+1} on L (the argument is symmetric for R), $[lo(v_i), hi(v_i)] \cap [lo(v_{i+1}, hi(v_{i+1})]$ is devoid of vertices, since in this case, the two guards cannot complete the discrete straight walk.

We define the following two functions on the vertices of L and R:

$$loV(v) = \min\{q \mid q \text{ vertex from } [lo(v), hi(v)]\}$$
$$hiV(v) = \max\{q \mid q \text{ vertex from } [lo(v), hi(v)]\}$$

Note that the values of loV and hiV must exist for both the vertices v_i and v_{i+1} by our assumption. Furthermore, since lo and hi are monotonic functions, so are loV and hiV. It was also shown in [IK] that $x \in [lo(y), hi(y)]$ if and only if $y \in [lo(x), hi(x)]$. We now argue that if $hiV(v_i) = x_j$ then $loV(v_{i+1}) = x_k$, then $k = j + 1$, since otherwise there would be a vertex between x_j and x_k, say x_l whose interval $[lo(x_l), hi(x_l)]$ would be devoid of vertices. Since every vertex is visible from its interval $[lo(v), hi(v)]$, we infer the following facts: v_i is visible from x_j, v_{i+1} is visible from x_{j+1}. Since v_i must be visible from v_{i+1} just as x_j is from x_{j+1}, either v_i or v_{i+1} must be visible from the entire edge joining x_j and x_{j+1}. This contradicts the assumption for the values of either $hiV(v_i)$ or $loV(v_{i+1})$.

The proof of the above theorem gives us a way for constructing the discrete straight walk (and consequently the hamiltonian triangulation), if one exists. The construction is described in the next section.

5 Algorithmic Applications

The previous section gives us a characterization for discretely straight walkable polygons. In this section we present efficient algorithms for several variants of the discretely straight walkability problem. Due to space constraints, we only give brief sketches in each case.

5.1 Walkability with respect to two given points

We first consider the problem of determining whether a given polygon P is discretely straight walkable with respect to two given points. Heffernan [H] presents a surprising linear-time algorithm for finding hi and lo values by ignoring redundant ray shots and computing only relevant information. The search structure used in his algorithm makes it more efficient. We use Heffernan's algorithm and compute the hi and lo values for each vertex. The only work left to check for discrete straight walkability with respect to the given s and t is to check whether any of the output intervals is devoid of vertices. Clearly this can be achieved in optimal linear time.

5.2 Recognizing Polygons with hamiltonian triangulations

This algorithm is similar to the one by Tseng and Lee [TL] to determine whether a polygon is straight walkable. The basic idea is to enumerate all deadlocks and all semi-wedges and all LR-visible pairs of the polygon P. All three can be computed in $O(n \log n)$ by simply computing all ray shots and scanning all the information after sorting. The scanning will help determine regions where the the points s and/or t cannot lie. If there is a portion that is not eliminated in the scanning, then clearly there exists a pair of points with respect to which the polygon has a hamiltonian triangulation. Even though the scanning is non-trivial it is straight-forward after sorting all the vertices and the ray shots.

5.3 Constructing a hamiltonian triangulation

The strategy for doing this is as follows. Theorem 2 shows that if P is discretely straight walkable (i.e., it has a hamiltonian triangulation), then loV and hiV values exist for each vertex of the chains L and R and that the intervals $[loV(v), hiV(v)]$ intersect for adjacent vertices on each of the two chains. Now the simple strategy for constructing the discrete straight walk is going to be: the guards move alternately (with guard one moving first) in such a way that when the stationary guard is at vertex v, the other guard moves until the vertex $hiV(v)$ is reached.

It is an easy step to now generate the triangulation. Every time the mobile guard moves to the next vertex, the edge connecting the stationary guard to the new position of the mobile guard is an edge of the triangulation. It also gives the next triangle visited by a hamiltonian path in the dual of the triangulation.

Clearly, this algorithm runs in optimal linear time, once the points s and t are known.

5.4 All-pairs discrete straight walkability

Several papers deal with problems of this nature [DHN, DHN2, TL]. Given a simple polygon this algorithm will output all pairs of points with respect to which the polygon is discrete straight walkable. As in the references mentioned, the output will be in the form of pairs of intervals. This algorithm follows the all-pairs straight walkability problem; the modification needed is to generate all semi-wedges, instead of all wedges. The extra processing that is needed is to inspect all ray shots that end on an edge and to identify all the semi-wedges that are not wedges. Details are omitted once again. Like the algorithm in [TL], this algorithm takes $O(n \log n)$ time.

6 Sequential Triangulations and Steiner Points

A hamiltonian triangulation is not a sequential triangulation when the hamiltonian path in its dual involves two consecutive *left* (or *right*) turns. Informally, a left (resp. right) turn in a hamiltonian path is easily understood by imagining the path as stepping on the triangles and then the next step must involve either a left or a right turn. A sequential triangulation is constrained to use a sequence of alternate left and right turns.

Given a triangulation, introducing a steiner point and retriangulating the triangle in which it is contained, changes the sequence of left and right turns in the associated hamiltonian. We state without proof that a hamiltonian triangulation can be transformed into a sequential triangulation by introducing at most $n/2$ steiner points and retriangulating the triangles. This is optimal in the sense that there are polygons with hamiltonian triangulations which need $n/2$ steiner points.

7 Open Problems

It is known [AHMS] that given any set of points in the plane there exists a hamiltonian triangulation of these points and that such a triangulation can be computed efficiently. The most interesting related problem that remains open is the following [AHMS]: given a set of points in 3-dimensional space, does there exist a *hamiltonian tetrahedralization* of the points? This problem is open even if the set of points are in convex, general position. Another variant of this problem is to consider hamiltonian triangulations in the skeleton of a tetrahedralization of the set of points. We can show that when the points are in convex, general position in 3-d, both the variants are equivalent. This problem seems to be related to variants of the *unfolding* problem for polytopes.

Other interesting problems relate to discrete versions of LR-visibility and their possible connections to discrete straight walkability.

References

[AHMS] E.M. Arkin, M. Held, J.S.B. Mitchell, S.S. Skienna, "Hamiltonian Triangulations for Fast Rendering," *Proc. of the 2nd ESA*, 1994.

[DHN] G. Das, P. J. Heffernan and G. Narasimhan, "LR-visibility in polygons," *Computational Geometry - Theory and Appln.*, **5**(1-2):37-57, (1997).

[DHN2] G. Das, P. J. Heffernan and G. Narasimhan, "Finding all weakly-visible chords of a polygon in linear time," *Nordic Journal of Computing*, **1** (1994), 433-457.

[ESV] F. Evans, S.S. Skienna, A. Varshney, "Optimizing Triangle Strips for Fast Rendering," *Proc. of VISUALIZATION*, 1996.

[H] P. J. Heffernan, "An optimal algorithm for the two-guard problem," *Proc. 9th Annual ACM Symp. on Computational Geometry*, (1993), pp. 348-358.

[IK] C. Icking and R. Klein, "The two guards problem," *Proc. 7th Annual ACM Symp. on Computational Geometry*, 1991, pp. 166-175.

[O87] J. O'Rourke, "Art Gallery Theorems and Algorithms," Oxford University Press, 1987.

[TL] L. H. Tseng and D. T. Lee, "Two-guard walkability of simple polygons," manuscript, 1993.