Hamiltonian Triangulations for Fast Rendering

Esther M. Arkin(1) * Martin Held(1),(2) †
Joseph S. B. Mitchell(1) ‡ Steven S. Skiena(3) §

(1)Department of Applied Mathematics and Statistics
State University of New York, Stony Brook, NY 11794-3600
(2)Institut für Computerwissenschaften
Paris-Lodron Universität Salzburg, A-5020 Salzburg, Austria
(3)Department of Computer Science
State University of New York, Stony Brook, NY 11794-4400

Abstract

High-performance rendering engines in computer graphics are often pipelined, and their speed is bounded by the rate at which triangulation data can be sent into the machine. To reduce the data rate, it is desirable to order the triangles so that consecutive triangles share a face, meaning that only one additional vertex need be transmitted to describe each triangle. Such an ordering exists if and only if the dual graph of the triangulation contains a Hamiltonian path.

In this paper, we consider several problems concerning triangulations with Hamiltonian duals. Specifically, we

- Show that *any* set of n points in general position in the plane has a Hamiltonian triangulation, and give an optimal $\Theta(n \log n)$ algorithm for constructing such a triangulation.
- Consider the special case of *sequential triangulations*, where the Hamiltonian cycle is implied, and prove that certain non-degenerate point sets in the plane do not admit a sequential triangulation. Further, we give efficient algorithms for testing whether a given triangulation of a point set or polygon is sequential.
- Show how to test whether a given polygon P has a Hamiltonian triangulation in time linear in the size of its visibility graph, and show that the problem is NP-complete for polygons with holes.
- Show how to add Steiner points to a given triangulation in order to create Hamiltonian triangulations which avoid narrow angles, thereby yielding guaranteed-quality Hamiltonian mesh generation.
- Give an encoding sequence for any triangulation whose length is at most 9/4 that of optimal.

*Partially supported by NSF Grants ECSE-8857642 and CCR-9204585.
†Partially supported by grants from Boeing Computer Services, and by NSF Grant DMS-9312098.
‡Partially supported by grants from Hughes Research Laboratories, Boeing Computer Services, Air Force Office of Scientific Research contract AFOSR-91-0328, and by NSF Grants ECSE-8857642 and CCR-9204585.
§Partially supported by NSF grant CCR-9109289 and ONR award 400x116yip01.

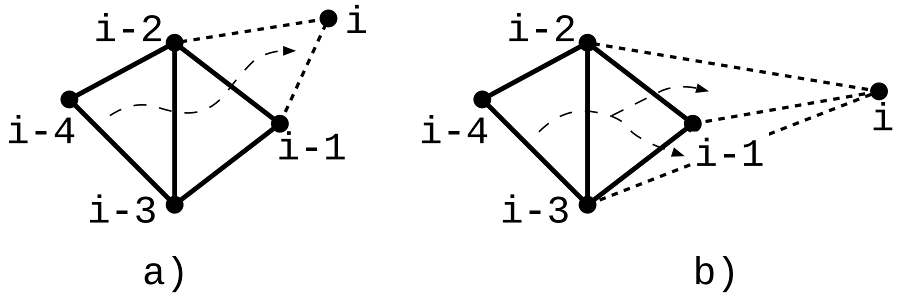

Figure 1: The new triangle need not be uniquely determined.

1 Introduction

The speed of high-performance rendering engines on triangular meshes in computer graphics can be bounded by the rate at which triangulation data is sent into the machine. Obviously, each triangle can be specified by three data points, but to reduce the data rate, it is desirable to order the triangles so that consecutive triangles share a face. Using such an ordering, only the incremental change of one vertex per triangle need be specified, potentially reducing the rendering time by a factor of three by avoiding redundant clipping and transformation computations.

A perfect ordering exists if and only if the dual graph of the triangulation contains a Hamiltonian path. We say that a triangulation is *Hamiltonian* if its dual graph contains a Hamiltonian path.

Even if a given triangulation of n vertices and m faces is Hamiltonian, the topology of the triangulation is not necessarily specified by the encoding sequence of vertices $v_0, \ldots, v_{m+1}$ defining the incremental changes, since, in general, v_i can form a triangle with either v_{i-1}, v_{i-2} or v_{i-1}, v_{i-3}. Figure 1a shows a case in which the next triangle is determined by the geometry, whereas both extensions are possible in Figure 1b.

Therefore, to completely specify the topology of the triangulation, we must specify the insertion order of the new vertices. At least two different models are currently in use, both assuming that all turns alternate from left to right. The Silicon Graphics triangular-mesh renderer *OPenGL* [18] demands that the user issue a *swaptmesh* call whenever the insertion order deviates from alternating left-right turns. In general, this topology can be transmitted at the cost of one extra bit per triangle. The *IGL* graphics library [3] expects triangulations as "vertex-strips", lists of vertices without turn specifications. To get two consecutive left or right turns the vertex must be sent twice, creating an empty triangle, which is discarded. For efficiency, it would be highly desirable to eliminate these extra bits or vertices by finding a path using an implied turn-order. We call a Hamiltonian triangulation with an implied left-right turn-order a *sequential* triangulation.

In this paper, we consider the problem of constructing Hamiltonian and sequential triangulations, towards the goal of increasing the efficiency of rendering in computer graphics. Two different classes of problems are of interest:

1. Given a point set or polygon, construct a Hamiltonian or sequential triangulation for it.

2. Given a triangulation T, find a Hamiltonian or sequential path for it, or failing that, either find a decomposition of T into a minimal number of Hamiltonian or sequential paths or add Steiner points such that a Hamiltonian or sequential path can be generated.

We are unaware of any previous results concerning Hamiltonian paths in the *dual* of triangulations. However, the graph theoretic properties of triangulations have been extensively studied. Dillencourt exhibited a point set whose Delaunay triangulation does not contain a Hamiltonian cycle [7], and showed that testing for the existence of such a cycle is NP-complete [9]. Hamiltonian cycles in Delaunay triangulations form the heart of the Connect-the-Dots heuristic [13] for extracting the shape of points in the plane. Further results on Hamiltonian cycles in Delaunay triangulations include [5, 8]; the length of the longest cycle in simple d-polytopes with n vertices is considered by [15].

Gray codes [11] are sequences of combinatorial objects which minimize the maximum incremental change between them. Hamiltonian triangulations can be considered as triangulations satisfying a certain "Gray code" property. Gray codes are known for various combinatorial objects, including subsets [20] and partitions [16].

In this paper, we consider several problems concerning triangulations with Hamiltonian duals. Specifically, we

- Show that *any* set of n points in general position in the plane has a Hamiltonian triangulation, and give an optimal $\Theta(n \log n)$ algorithm for constructing such a triangulation. We have implemented and tested our algorithm.
- Prove that certain non-degenerate point sets in the plane do not admit a sequential triangulation, and give efficient algorithms for testing whether a given triangulation or polygon is sequential.
- Show how to test whether a given n-gon P has a Hamiltonian triangulation in time linear in the size of its visibility graph, and show that the problem is NP-complete for polygons with holes.
- Show how to add Steiner points to a given triangulation in order to create Hamiltonian triangulations which avoid narrow angles, thereby yielding guaranteed-quality Hamiltonian mesh generation.
- Give an encoding sequence for any triangulation whose length is at most 9/4 that of optimal.

2 Hamiltonian Triangulations of Polygons

In this section, we consider the problem of constructing Hamiltonian triangulations of simple polygons. The dual of any triangulation of a polygon is a tree, where the leaf nodes of the dual represent ears of the polygon, triangles with two of its three faces defined by the boundary of the polygon. A triangulation is Hamiltonian if and only if it contains exactly two ears – i.e., if and only if the dual graph is a path.

Convex polygons trivially have Hamiltonian triangulations – simply triangulate with all chords incident upon a single vertex, as shown in Figure 2. However, the nonconvex simple polygon in Figure 3 has no Hamiltonian triangulation. We present an efficient algorithm for finding a Hamiltonian triangulation of a polygon, if it exists. We also consider the problem of adding few Steiner points to polygons so as to permit Hamiltonian triangulations.

Deciding whether a polygon has a Hamiltonian triangulation is related to the concepts of LR-visibility and straight walkability. A polygon P is *LR-visible* if P can be partitioned into two

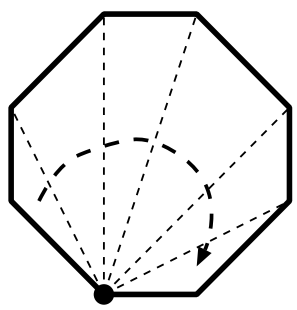

Figure 2: A Hamiltonian triangulation of a convex polygon

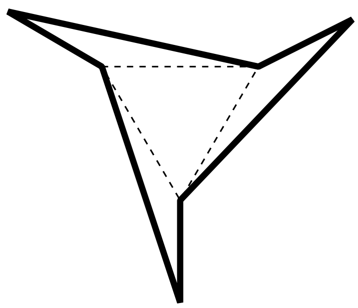

Figure 3: Straight walkability does not imply a Hamiltonian triangulation.

subchains, called L and R, such that every point of L can be seen by at least one point on R and vice versa. Das, Heffernan, and Narasimhan [6] show how to test for LR-visibility in linear time. Any polygon with a Hamiltonian triangulation is clearly LR-visible, but Figure 3 shows the converse is not true. A polygon P is *straight walkable* if it has vertices s and g which partition P into two chains, such that two "guards" walking from s to g on different chains without backtracking are mutually visible at all times. Tseng and Lee [19] give an $O(n \log n)$ time algorithm to find all pairs of points s and g for which an n sided polygon is straight walkable. Any polygon that is straight walkable is clearly LR-visible, and any polygon with a Hamiltonian triangulation is clearly straight walkable. However, Figure 3 shows a straight walkable polygon with a unique triangulation, which is not Hamiltonian. (Note that this example polygon is also monotone and star-shaped.)

We note that polygons which are monotone and star-shaped need not admit a Hamiltonian triangulation. Again, Figure 3 serves as a counterexample.

We say that a polygon is *discretely straight walkable* if it has vertices s and g which partition P into two chains, such that two "guards" can walk from s to g on different chains without backtracking, where only one of the guards is allowed to walk at a time, while the other guard rests at a vertex, and the two guards are mutually visible whenever both guards are at vertices of the polygon. Clearly, a polygon is discretely straight walkable if and only if it has a Hamiltonian triangulation.

Theorem 2.1 *The problem of testing whether a given simple n-gon P has a Hamiltonian triangulation can be solved in $O(|E|)$ time, where $|E|$ is the number of visibility graph edges in the polygon.*

Proof. We use dynamic programming. Assume that the vertices of P are numbered in clockwise order. Any visible chord (i, j) of P represents a potential triangulation edge, and partitions P into two subpolygons, one to the left of (i, j) whose vertices are $(i, i+1, \ldots, j)$, and one to the right of (i, j) whose vertices are $(j, j+1, \ldots, i)$, where all additions and subtractions are modulo n. Let $D[i, j]$ be a Boolean function on a pair of vertices i and j which is true if and only if the subpolygon to the left of (i, j) has a Hamiltonian triangulation ending with (i, j). Note that it suffices to define $D[i, j]$ for all pairs of vertices i, j which are mutually visible.

Observe that for all $1 \leq i \leq n$, $D[i, i+2]$ is true iff $(i, i+2)$ is a chord of P. The dynamic programming recursion for $D[i, j]$ is

$$D[i, j] = \{D[i, j-1] \text{ and } (i, j-1) \text{ visible}\} \text{ or } \{D[i+1, j] \text{ and } (i+1, j) \text{ visible}\}.$$

P has a Hamiltonian triangulation if and only if there exists a pair of vertices i, j such that both $D[i, j]$ and $D[j, i]$ are true. Once the visibility graph of P has been constructed in $O(|E|)$ time using Hershberger's algorithm [12], we maintain the visibility graph edges in buckets numbered $k = 1, 2, \ldots, n-1$ for $i - j = k$, and process them in order of increasing k. (Edges of the polygon are *not* considered to be edges of the visibility graph.) It is easy to see that each of the $O(|E|)$ computations of the recurrence takes constant time, since Hershberger's algorithm yields a data structure in which the visible edges from each vertex are sorted in order around the polygon. To answer whether $(i, j-1)$ is visible, we need only ask whether the previous (clockwise) visible vertex from i before j is $j-1$. Similarly we can decide in constant time whether $(i+1, j)$ is visible. □

Unfortunately, the problem becomes intractible for polygons with holes.

Theorem 2.2 *Given a polygon with (polygonal) holes, it is NP-complete to determine whether there exists a triangulation of the interior whose dual graph is Hamiltonian.*

Proof. We use a reduction from the known NP-complete problem of determining whether a planar cubic graph is Hamiltonian [10]. To avoid confusion, we refer to graph vertices as *nodes*, polygon vertices as *vertices*, graph edges as *arcs*, and polygon or hole edges as *edges*.

Given a straight line plane drawing of an arbitrary planar cubic graph G, we construct a polygon with holes corresponding to the interior faces of the planar graph. Each arc of G is mapped to a narrow "V"-shaped tunnel. Thus, for every arc we have 4 edges that are part of the boundary of the polygon or its holes. Each node in G will have three corresponding vertices, which we refer to as *node-vertices*. For every tunnel there are thus two pairs of node-vertices at the "mouths" of the tunnel. Every arc of G has two additional vertices at the bend of the "V", which we call *arc-vertices*. See Figure 4 for an illustration, where vertices A–F are node-vertices, B–E are at the mouths of the tunnel, and vertices G and H are arc-vertices.

It is not difficult to see that we can construct such a polygon with holes so that each of the two arc-vertices are visible to both sets of 3 node-vertices corresponding to nodes incident to that arc. Furthermore, no node-vertex is visible to another node-vertex unless they correspond to the same node in the graph. This completes the construction of a polygon with holes P, which can clearly be done in polynomial time.

Note that any triangulation of P must use all diagonals connecting pairs of arc-vertices, for otherwise certain triangles would have interior angles larger that 180 degrees. We refer to these diagonals as *forced* diagonals; in Figure 5, forced diagonals are shown by solid lines whereas the other diagonals are depicted using dotted lines. Thus every triangle in the triangulation is formed

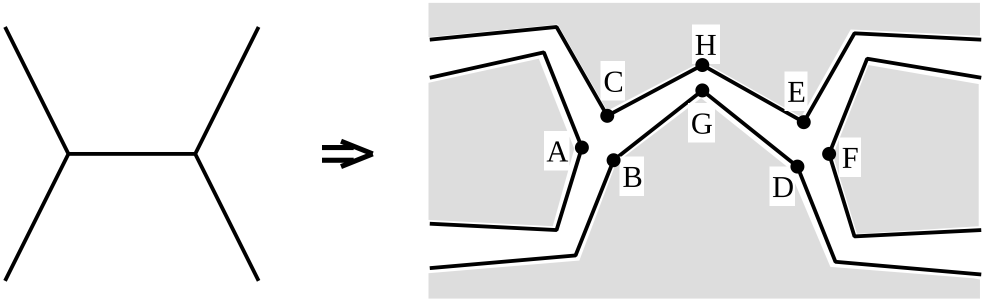

Figure 4: Transforming a graph to a polygon with holes.

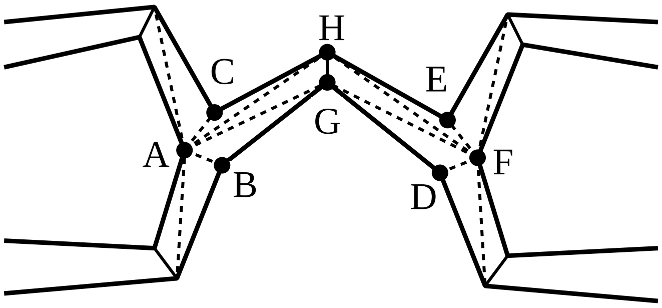

Figure 5: Hamiltonian triangulations imply Hamiltonian paths.

by 3 node-vertices, or 2 node-vertices and 1 arc-vertex, or 1 node-vertex and 2 arc-vertices, where the node-vertices correspond to the same node in the graph, and the arc-vertices correspond to a one of the arcs incident to that node.

We now show that if G has a Hamiltonian path (circuit) then there must be a triangulation whose dual contains a Hamiltonian path (circuit). After inserting all forced diagonals, we complete the triangulation as follows. For arcs in the Hamiltonian path, we triangulate the corresponding tunnel into 4 triangles, by connecting each pair of node-vertices at the mouth of the tunnel, obtaining 2 rectangles that are triangulated by one additional (arbitrary) diagonal for each. For arcs not in the Hamiltonian path, we have 6 triangles obtained by connecting the two arc-vertices to both node-vertices (whose corresponding nodes are incident to the corresponding edge) that are not at the mouth of the tunnel. In Figure 5 the path does not pass through the tunnel corresponding to the horizontal edge. The Hamiltonian path in G has a corresponding Hamiltonian path in the dual of the triangulation described, which goes through the four triangles of the arcs in the graph-path, and before proceeding to the next arc in the graph-path goes through 3 of the 6 triangles of the arc not in the graph-path.

To show the converse, namely that if there exists a Hamiltonian triangulation in P then G is Hamiltonian, we recall that the diagonals connecting pairs of arc-vertices are forced in any triangulation. The Hamiltonian path in the dual to the triangulation crosses each such diagonal at most once. The arcs corresponding to forced diagonals that the path crosses once form a Hamiltonian path in the original graph. □

While not all polygons admit a Hamiltonian triangulation, it turns out that any polygon can be made Hamiltonian by adding Steiner points. We have two results – the first demonstrating that

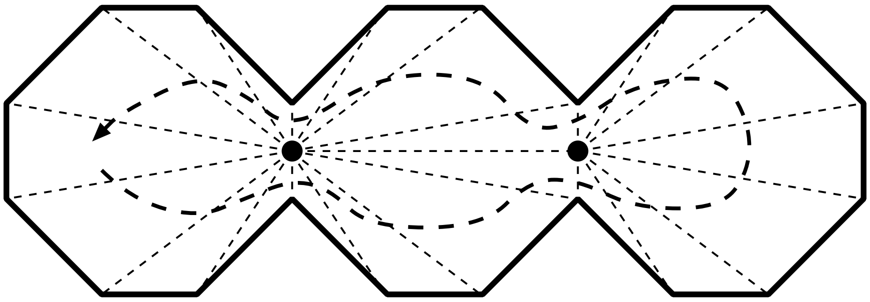

Figure 6: Placing Steiner points to create a Hamiltonian triangulation.

few Steiner points always suffice, the second showing that a modest number of Steiner points can result in a Hamiltonian triangulation that avoids slivery triangles.

Theorem 2.3 *Let P be a simple n-gon which is partitioned into $k > 1$ convex regions in its minimum convex decomposition. Then a Hamiltonian cycle (resp. path) triangulation can be constructed for P using $k-1$ (resp. $k-2$) Steiner points in $O(n)$ time. Furthermore, there exist simple n-gons that require $k-1$ (resp. $k-2$) Steiner points for a Hamiltonian triangulation.*

Proof. First, we show that $k-1$ Steiner points suffice for a Hamiltonian cycle triangulation. If $k = 2$ then a fan-shaped triangulation can be constructed by placing a Steiner point onto the common boundary of the two convex subareas and by connecting every vertex of the polygon to the Steiner point. If $k > 2$ we consider a spanning tree of the decomposition into convex subareas. The subarea corresponding to the root of the tree and one of its adjacent subareas, the so-called root subareas, are handled as in the case $k = 2$. For every subarea adjacent to the root subareas, the same scheme is applied by placing a Steiner point onto the common boundary of this subarea and a root subarea. By traversing the entire tree a Hamiltonian cycle triangulation is generated, thereby spending exactly $k-1$ Steiner points. In Figure 6 the left-most and the middle convex subarea were chosen root subareas.

A Hamiltonian path triangulation using $k-2$ Steiner points can be generated by moving the Steiner point spent on triangulating the root subareas towards one of the two vertices of their common boundary, thus saving one Steiner point. The remaining areas are processed as in the case of a Hamiltonian cycle triangulation.

To show necessity, consider the rectilinear comb in Figure 7. Any Hamiltonian triangulation requires the claimed minimum number of Steiner points for either Hamiltonian paths or cycles. □

We note that exactly the same proof also applies to polygons with polygonal holes. Thus, Theorem 2.3 is also valid for multiply-connected planar shapes. However, computing a minimum convex decomposition is NP-hard for multiply-connected planar shapes, cf. [14]. Therefore, we have to content ourselves with approximations of the minimum decomposition, thereby losing the worst-case optimality of our construction.

Theorem 2.4 *Every triangulation T containing k triangles can be converted into a Hamiltonian cycle triangulation by adding k Steiner points such that the minimum angle of the Hamiltonian triangulation is not less than half of the minimum angle of the original triangulation.*

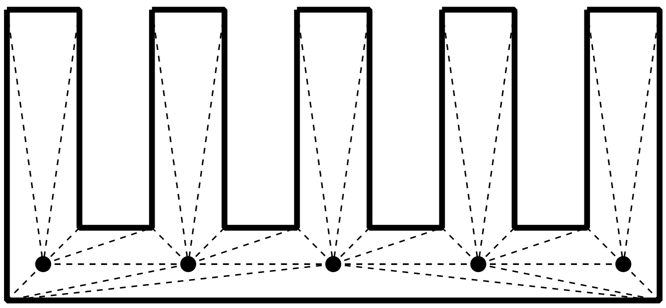

Figure 7: A polygon requiring many Steiner points for a Hamiltonian triangulation.

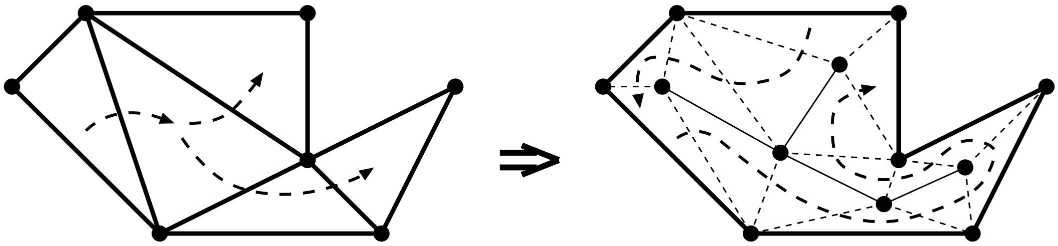

Figure 8: Retriangulation after adding Steiner points.

Proof. For every triangle of the original triangulation T, we place a Steiner point at the center of the triangle's largest inscribed circle and connect it to the vertices of the triangle. Now consider a spanning tree of the dual graph of T, and flip all edges belonging to T which are crossed by the spanning tree. This modification yields a Hamiltonian triangulation. In Figure 8, the edges of T are depicted by thick solid lines, while the results of edge flips are depicted by thin solid lines and the rest of the new triangulation edges are dotted.

Next we show that all edges of T which are to be flipped can actually be flipped. Consider Figure 9, where the line segment AB denotes an edge of T which is to be flipped, and C and D are the centers of the inscribed circles of the original triangles sharing the edge AB. By construction, the angles $\angle(C, A, B)$ and $\angle(B, A, D)$ are half of the corresponding angles of the original triangles at A, and the same argument holds for the angles at B. Thus, $\angle(C, A, B) + \angle(A, B, C) < \pi/2$. Consequently, $\angle(B, C, A) > \pi/2$, and C (and D) lie within the circle which has AB as its diameter. We conclude that (A, D, B, C) forms a convex quadrilateral and that AB can be flipped, thereby replacing it by CD.

It remains to show that any angle of the Hamiltonian triangulation is no smaller than half that of the smallest angle of the original triangulation. The only angles which must be examined are those around the Steiner points. WLOG, we consider the angle $\angle(C, D, B)$ and show that $\angle(C, D, B) \geq \angle(C, A, B)$; similar arguments hold for the other angles around C and D. Recall that C and D lie inside the circle with diameter AB. Let E be the intersection of the line through A and C with this circle, and let F be the intersection of the line through D and E with this circle, as shown in Figure 9. From the construction we get the sequence of inequalities

$$\angle(C, D, B) \geq \angle(E, D, B) \geq \angle(E, F, B).$$

The proof is finished by observing that $\angle(E, F, B) = \angle(E, A, B)$. $\square$

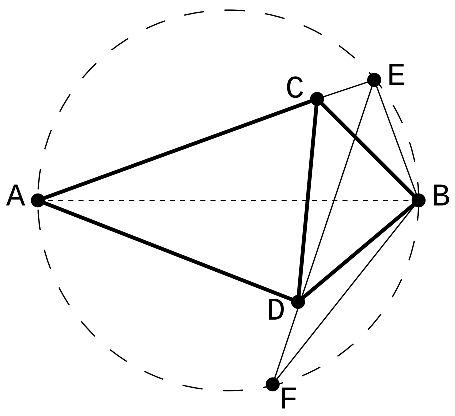

Figure 9: No small angle is introduced with Steiner points.

We note that this result enables us to postprocess any of the known "guaranteed-quality" triangulations and to convert them into a Hamiltonian triangulation, thereby roughly preserving the features of these triangulations. For instance, we may want to minimize the maximum angle or maximize the minimum angle. We refer to Bern and Eppstein [2] for an extensive review of techniques for generating guaranteed-quality triangulations. Of course, such a postprocessing enlarges the triangulation, which works against our goal of minimizing the data to transmit.

In "conformal" mesh-generation, some of the edges of the original triangulation may not be flipped or removed. In this case, a Hamiltonian triangulation can be generated by placing additional Steiner points on the midpoints of the conformal edges. By observing that the center of the largest inscribed circle of a triangle is constrained to the interior of a similar subtriangle formed by the midpoints of the triangle's edges it can be shown that the introduction of these additional Steiner points does not yield any new angle smaller than one of the already existing angles. Alternatively, care can be taken during the generation of the spanning tree of the dual graph of the original triangulation that no edge which may not be flipped needs to be crossed. We omit details in this abstract.

3 Hamiltonian Triangulations of Point Sets

We now consider the question of whether or not a set of points has a Hamiltonian triangulation. We begin by observing that the Delaunay triangulation is not necessarily Hamiltonian; see Figure 10. We will show that, surprisingly, *any* non-degenerate point set in the plane has a Hamiltonian triangulation, and give an elegant and efficient algorithm for constructing it. In fact, we have computational experience with algorithms that we have implemented to compute Hamiltonian triangulations; we show an example output in Figure 11. Our triangulation algorithm proves to be similar to one of Avis and ElGindy [1] (also Yvinec [21]), although they do not consider the question of finding Hamiltonian cycles in the dual graph.

Theorem 3.1 *Let S be a set of n points in the plane. If S is convex, it has a Hamiltonian path triangulation. Otherwise, S has a Hamiltonian cycle triangulation, and such a triangulation can be constructed in $O(n \log n)$ time.*

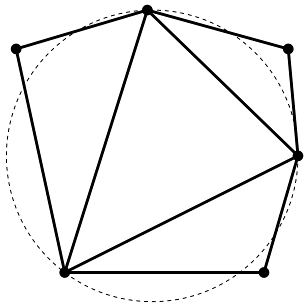

Figure 10: The Delaunay triangulation is not necessarily Hamiltonian.

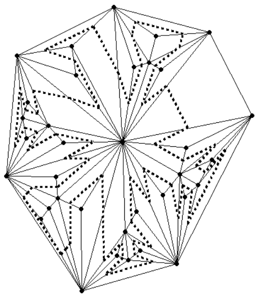

Figure 11: A Hamiltonian triangulation of a point set.

Proof. For simplicity, we will assume that S is in general position (no 3 collinear points); J. Mittleman has observed that this assumption is not necessary. If S is convex, then it clearly has a Hamiltonian path triangulation – simply add all chords incident upon any single point to the convex hull of S, $conv(S)$. Therefore, we may assume that there is at least one point v in the interior of $conv(S)$.

Adding chords from v to each vertex of $conv(S)$ yields a triangulation, which has a Hamiltonian cycle in its dual. Each of the remaining points of S lies within a face of this triangulation.

We will now add any additional interior points in an incremental fashion. Because of the non-degeneracy assumption, each point lies in the interior of one face of the current triangulation. Adding three triangulation edges from the new point to the three vertices of its enclosing triangle T divides the triangle into three new triangles. One of the new triangles will contain the edge crossed entering T by the Hamiltonian cycle, while another will contain the exit edge of T. As shown in Figure 12, three triangles can be connected to respect this ordering and preserve the Hamiltonian cycle.

To prove the desired time bound, we must identify a good splitter at each step. Avis and ElGindy [1] prove that in any set of n points within a triangle, there always exists a splitter such that none of the three resulting triangles contains more than $2n/3$ points. Further, this point can be found in $O(n)$ time. This yields a recursion tree of logarithmic height, with a linear amount of time spent partitioning at each level. Thus $\Theta(n \log n)$ time sufficies to construct a Hamiltonian cycle triangulation of n points. $\Box$

The original deterministic algorithm by Avis and ElGindy does not lead to an easy implementation, as it relies on linear-time median finding in one dimension. Therefore, it is interesting to

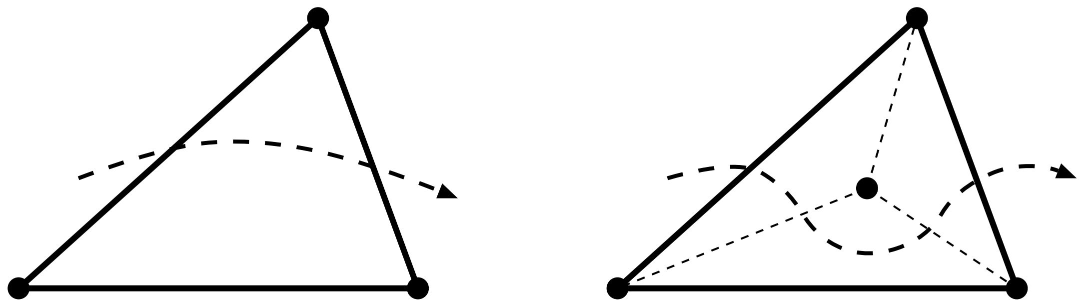

Figure 12: Updating a Hamiltonian triangulation with a new interior point.

note that a simple randomized algorithm achieves the same expected time bound:

Lemma 3.2 *Using a random splitter in the algorithm of Theorem 3.1 produces a Hamiltonian triangulation in expected $O(n \log n)$ time.*

Proof. We employ the backward analysis technique [17]. Let $S = \{p_1, p_2, \ldots p_n\}$ be a set of n points inside a triangle. We assume that the points of S have been permuted, and are given in random order. Our algorithm has n iterations. At the beginning of the rth iteration the point p_r is selected to be the next "splitter" point. The points $\{p_r, p_{r+1}, \ldots, p_n\}$ are inductively known to be in one of the $1 + 2(r-1)$ existing triangles. We located (in constant time) the triangle T_k containing p_r, whose endpoints are p_i, p_j, p_l and $i, j, l < r$. Remove T_k from the list of triangles, and create three new triangles instead (with endpoints p_i, p_j, p_l, p_r). All that remains is to split the subset of points of $\{p_{r+1}, p_{r+2}, \ldots, p_n\}$ that were in the triangle T_k among the three newly created triangles. For each point, this is a constant time operation. We now analyze the expected number of times that we must decide to which of three triangles does a given point belong.

Consider an arbitrary point $p \in S$; we ask how many times do we have to answer the question to which of the 3 new triangles it belongs. At a particular iteration r, if $p \in \{p_1, p_2, \ldots, p_r\}$, then certainly the question does not arise. We thus assume that $p \notin \{p_1, p_2, \ldots, p_r\}$, which means that p is interior to some current triangle T_i. In this case, the question arose only if T_i was a "new" triangle, namely it did not exist in iteration $r-1$. However, that can happen only if one of the three endpoints of T_i is p_r. Using backwards analysis we can say that any one of the points already used as splitters up to now had the same probability of being p_r, and thus this probability is $3/r$. In other words, the expected number of times that we must answer the question for a point p in round r is $3/r$, and summing over all the rounds we get $O(\log n)$ per point, resulting in the desired $O(n \log n)$ expected running time. □

4 Sequential Triangulations

We say that a triangulation is *sequential* if its dual graph contains a Hamiltonian cycle (resp., path) such that no three triangulation edges consecutively crossed by the Hamiltonian cycle (resp., path) are incident upon the same vertex of the triangulation. This implies that all turns alternate left-right, so sequential triangulations can be described without the extra bit-per-face needed to specify a general Hamiltonian triangulation.

Sequential triangulations represent a significant restriction. For example, convex polygons have an exponential number of Hamiltonian triangulations but have only a linear number of sequential triangulations. Also, point sets that admit Hamiltonian cycle triangulations may not have sequential cycle triangulations, as illustrated by a triangle with an interior point. The most tantilizing question is whether sequential path triangulations always exist for any non-degenerate point set, as do Hamiltonian path triangulations.

After an extensive computer search (of over 300,000 randomly generated configurations of point sets), followed by analysis, we have resolved this question in the negative:

Theorem 4.1 *For any $n \geq 9$, there exists a set of n points in general position that does not admit a sequential triangulation.*

Proof. We will show that the set of points depicted in Figure 13 admits no sequential triangulation. The defining property of this point set is that the points D, E, F, I are on the right side of the oriented line (A, B) through A and B, E, F, G, H are on the right side of (B, C), and G, H, I, D are on the right side of (C, A).

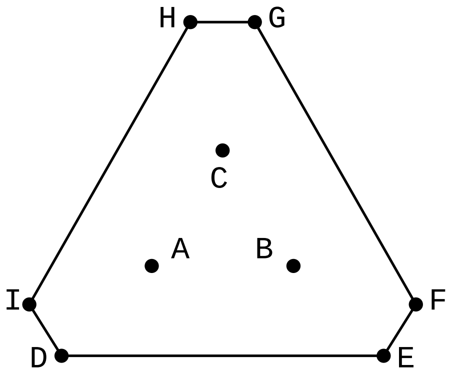

Figure 13: A point set with no sequential triangulation

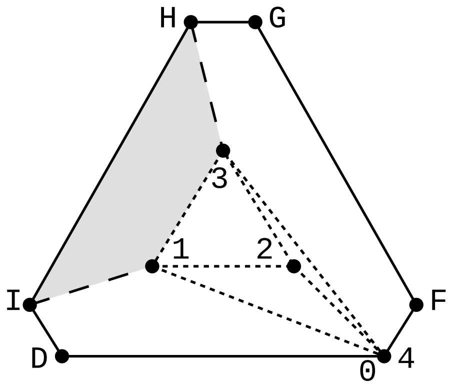

Figure 14: Case 1

Due to the geometry, any triangulation of the points must contain the triangle (A, B, C). Below, we will assign numbers to the individual points in the order they can be visited by a sequential triangulation, where the points are visited in increasing order. Due to symmetry, we may w.l.o.g. assign the numbers $1, 2, 3$ to the points A, B, C. Once the numbering of A, B, C has been fixed, there exist four candidates which can be assigned the number 4: E, F, G, H.

Figure 14 depicts the situation where we have chosen E as the fourth point. If a point is to be labeled 0, then E is the only suitable candidate. However, no matter which points are numbered 5 (and -1), we are not able to cover the shaded area with triangles. The same argument holds if we label F as 4 instead of E.

Now assume that point G is labeled 4, as in Figure 15. We could assign the number 0 to either D, E, F, I. However, such a labeling makes it impossible to cover the shaded area by a triangle. Thus, if we assign the number 4 to G, the sequential triangulation must start at triangle (A, B, C). However, in this case we are forced to assign 5 and 6 to A and H, as depicted in Figure 16. Again, it can be seen that the shaded area cannot be covered by triangles, no matter which point is labeled

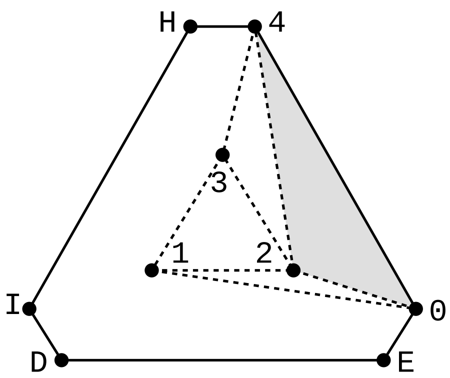

Figure 15: Case 2

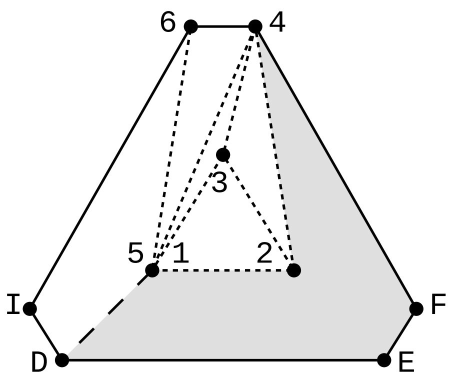

Figure 16: Case 3

7. The same argument holds if we assign 4 to H instead of G, which completes our proof for the case of $n \geq 9$.

The configuration given in Figure 13 can be generalized to $n = 9 + k$ points by adding k points such that

- all k points lie on the convex hull; i.e., the convex hull now has $6 + k$ points.
- no additional point is placed on the hull arcs $E - F$, $G - H$ and $I - D$, where the hull arcs are specified according to a CCW traversal of the convex hull.

The same argument sufficies to show that no sequential triangulation exists. □

Lemma 4.2 *Let S be a set of points (in general position). If at most two points of S lie strictly in the interior of the convex hull of S then S has a sequential triangulation.*

Proof. Let $p_1, \ldots, p_n$ be the vertices of the convex hull of S (arranged in CCW order); in the sequel all index arithmetic is done modulo n.

The proof is trivial if S is in convex position. If there exists exactly one point q_1 which is interior to the convex hull of S then we determine the hull edge (p_i, p_{i+1}) which is intersected by the line through p_1 and q_1. Obviously, q_1 is in the inside of the triangle $\Delta(p_1, p_i, p_{i+1})$. After triangulating $\Delta(p_1, p_i, p_{i+1})$ with respect to q_1 a sequential triangulation of S is easily obtained.

Now assume that there are two points q_1, q_2 which are interior to the convex hull of S. First we determine the two hull edges (p_i, p_{i+1}) and (p_j, p_{j+1}) which are intersected by the line through q_1 and q_2. W.l.o.g. we assume that q_2 is closer to (p_i, p_{i+1}) than q_1. Note that $\Delta(q_1, p_i, p_{i+1})$ contains q_2 and that $\Delta(q_2, p_j, p_{j+1})$ contains q_1. The rest of the proof is done by a case analysis.

Case 1: (p_i, p_{i+1}) and (p_j, p_{j+1}) are two consecutive edges of the convex hull. W.l.o.g. we assume that $j + 1 = i$, cf. Fig. 17a, where filled circles denote the points of the convex hull, and unfilled circles denote q_1, q_2. Edges of the convex hull are depicted by solid lines whereas edges of the triangulation are depicted by dashed lines. Fig. 17b shows a numbering of the points according to a sequential triangulation. If $n = 3$ then we are done; otherwise this triangulation is extended beyond the edge (p_{i+1}, p_{i-1}) in order to cover all points of S, and we are also done. Thus, for the remaining cases we may safely assume that $i \neq j + 1$ and $i + 1 \neq j$, which implies $n \geq 4$.

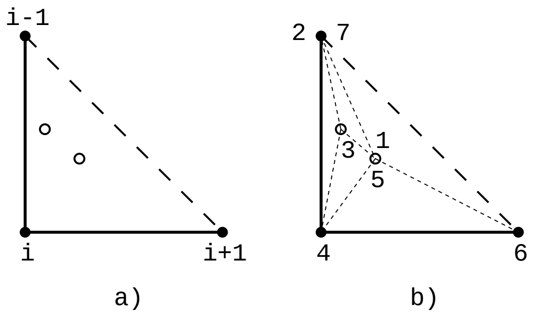

Figure 17: Proof of Lemma 4.2: Case 1

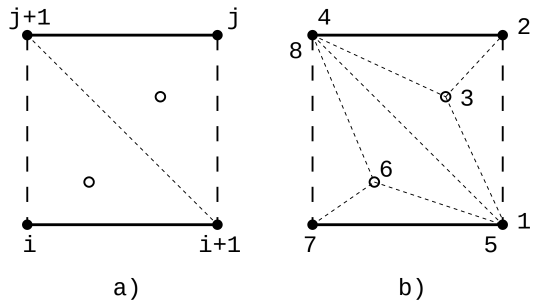

Figure 18: Proof of Lemma 4.2: Case 2

Case 2: $q_1 \in \Delta(p_{i+1}, p_j, p_{j+1})$ and $q_2 \in \Delta(p_i, p_{i+1}, p_{j+1})$, cf. Fig. 18a, or $q_1 \in \Delta(p_i, p_j, p_{j+1})$ and $q_2 \in \Delta(p_i, p_{i+1}, p_j)$. After triangulating both triangles according to q_1 and q_2 it is straightforward to number all points such that a sequential triangulation is obtained, cf. Fig. 18b. Thus, for the rest of the proof we may assume that any of these four triangles either contains both q_1 and q_2 or none of them. W.l.o.g. we assume that both q_1 and q_2 lie in $\Delta(p_i, p_{i+1}, p_{j+1})$.

Case 3: $j + 1 = i - 1$. If $q_1, q_2 \in \Delta(p_i, p_{i+1}, p_{j+1}) \cap \Delta(p_i, p_j, p_{j+1})$, cf. Fig. 19a, then a suitable numbering of the points is given in Fig. 19b. If $q_1, q_2 \in \Delta(p_i, p_{i+1}, p_{j+1}) \cap \Delta(p_i, p_{i+1}, p_j)$, then a similar numbering can be accomplished.

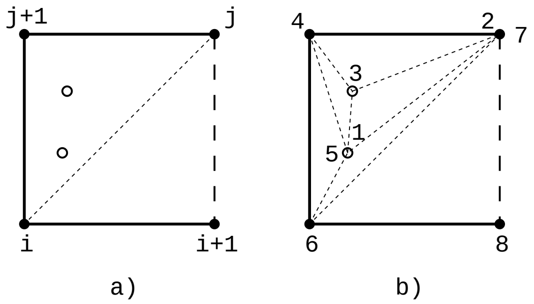

Figure 19: Proof of Lemma 4.2: Case 3

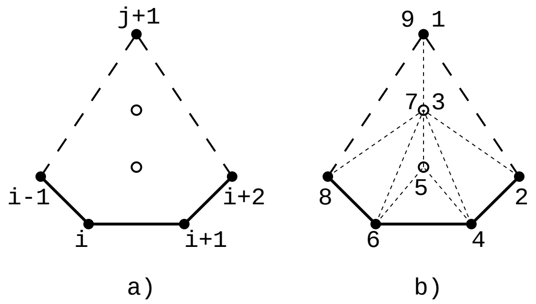

Figure 20: Proof of Lemma 4.2: Case 4

Case 4: none of the first three cases applies. Note that in this case $n > 4$ holds, and that $i + 2 \neq j + 1$ and $i - 1 \neq j + 1$. Thus, we are dealing with a set-up of the points as depicted in Fig. 20a. In Fig. 20b, we give a suitable numbering of the points, which completes this case and the entire proof. □

We may ask the same questions for sequential triangulations that we have asked for Hamiltonian triangulations: For a given point set or polygon, does a sequential triangulation exist and, if so, how can we compute one? While testing whether a triangulation is Hamiltonian is hard (see Section 5), a fast algorithm exists to test whether it is sequential:

Theorem 4.3 *Testing whether a given triangulation on n points is sequential can be done in $O(n)$ time.*

Proof. Observe that from a specific ordering of three consecutive vertices of a sequential triangulation, the potential predecessor and successor vertices are completely defined. Therefore, any given triangle has only six possible orientations.

Select an arbitrary triangle as a starting point, and for each orientation compute the sequence of predecessors until it self-intersects, ie. a triangle occurs twice among the predecessors. After constructing a similar sequence among the successors, by concatenating the two sequences together we have a chain of length at most $2m$, where m is the number of triangles. This orientation of the triangle is involved in a sequential triangulation iff there is a substring of the chain consisting of m triangles without repetition.

Clearly, each of the six complete chains can be constructed in linear time. To test for an appropriate substring, we sweep from left to right, maintaining an array of how many times each triangle has occurred in the last m elements, and a counter of how many distinct triangles have been covered in the last m elements. For each new element, we delete one from the count of the triangle leaving the length-m window, decrement the distinct triangle counter if this has gone to zero, increment the count for the new triangle, and increment the distinct triangle counter if this has changed to one. We report a sequential triangulation iff the distinct count has gone to m. Each of the $O(n)$ updates can be performed in constant time. $\square$

Testing whether a polygon has a sequential triangulation can be done within the same time bound as testing whether it has a Hamiltonian triangulation:

Theorem 4.4 *The problem of testing whether a given polygon P has a sequential Hamiltonian triangulation can be solved in $O(|E|)$ time, where $|E|$ is the number of visibility graph edges for P.*

Proof. Assume that the vertices of the polygon are numbered in clockwise order. Note that once we pick a vertex k of the polygon to be numbered 1, there are only two choices for the vertices to be numbered 2 and 3, namely $k-1$ and $k+1$, after which the remaining numbering is unique. We thus restrict our attention to finding a suitable choice for the first vertex, and which of its neighbors will be numbered 2.

Observe that each visibility graph edge (i, j) can be an edge in a sequential Hamiltonian triangulation for at most two vertices of the subchain $(i, i+1, \ldots, j)$ being assigned the number 1 (all additions and subtractions are modulo n). When $i+j$ is odd, the two vertices are $\lfloor (i+j)/2 \rfloor$ and $\lceil (i+j)/2 \rceil$, denoted type L and R, respectively. When $i+j$ is even, this vertex is given by $(i+j)/2$, and we refer to it as both a type L and R vertex. We initialize our algorithm by setting the number of 'votes of confidence' of type L and R for each vertex to 0. For each visibility graph edge, we increase the number of type L and R votes for the appropriate one or two vertices, since each such edge increases two counts. Finally, a vertex can be numbered 1 in a sequential Hamiltonian triangulation if and only if it received exactly $n-3$ such type L (or R) votes, where $n-3$ is the number of diagonals in a triangulation of an n-gon. $\square$

5 Path Coverings and Partitionings of Triangulations

We have seen that any set of points admits a Hamiltonian triangulation (Theorem 3.1) and that, by adding Steiner points to an existing triangulation, we are always able to make a triangulation Hamiltonian (Theorem 2.4). However, in some applications, we are required to work with a given triangulation and are not free to add Steiner points. Hence, we are faced with the problem of ordering the faces of an *existing* triangulation to minimize the transmission rate.

The problem of testing whether a given triangulation is Hamiltonian is easily seen to be NP-complete, by reduction from Hamiltonian circuit/path in planar cubic graphs [10]: Given an instance of a planar cubic graph, embed it in the plane, take its dual, and then redraw the dual using straight line segments. This can be done so that the exterior face is convex; hence, we obtain a triangulation. Therefore, the problem of finding the minimum path cover or minimum path partition is also hard. The problem is easy for polygons, however:

Lemma 5.1 *A minimum path partition of a simple n-gon triangulation can be constructed in $O(n)$ time.*

Proof. The dual graph of a polygon triangulation is always a tree; hence, this problem reduces to finding a minimum path cover or partition of a tree. Clearly, half the number of odd-degree vertices represents a lower bound on the number of paths in a partition or cover. This bound can be realized by a greedy strategy – any path that originates and ends on odd-degree vertices can be deleted to leave a graph with two fewer odd-degree vertices. □

For our motivating application, we seek a description of a triangulation that minimizes the encoding length. The Silicon Graphics triangular-mesh renderer *OPenGL* mentioned in the introduction demands a sequential representation; i.e., all turns alternate from left to right. To get two consecutive left or right turns the vertex must be sent twice, creating an empty triangle, which is discarded, since there is no mechanism to specify turn bits.

By specifying empty triangles and redrawing previously rendered triangles, every triangulation has a legal encoding sequence of vertices. To show this, consider a spanning tree of the dual of the triangulation, and a depth-first traversal of this tree, giving a walk which visits each of the m triangles twice. Empty triangles can be used to ensure that turns alternate left and right. Our goal is to build as short an encoding sequence as possible for a given triangulation.

An ear of a point-set triangulation is a triangle defined by two convex hull edges. Ears lead to long encoding sequences, for once an ear has been entered, two vertices must be repeated in order to leave the ear. We say a triangulation is *deaf* if it has no ears.

Lemma 5.2 *For any encoding sequence of a deaf triangulation which visits m' non-empty triangles, there is an encoding sequence that has at most $m'/2$ empty triangles.*

Theorem 5.3 *Let S_{opt} be an encoding sequence of minimum length for a given triangulation T on m triangles. An encoding sequence for T of length at most $(9/4) \cdot |S_{opt}| + O(1)$ can be constructed in $O(n^3)$ time.*

Proof. Our algorithm uses the Christofides heuristic [4] for traveling salesman problems satisfying the triangle inequality. The vertices of the graph will be the triangles in T – the weight of an edge will be the shortest path length in the dual between the pair of triangles. Any solution to this TSP defines a walk which visits each triangle in T – the Christofides solution length, m', will be within 3/2 of optimal (and clearly, $m \leq m' \leq 2m$).

A Christofides walk may contain repeated left or right turns, which require empty triangles to turn into a sequential walk. Assuming T is a deaf triangulation, by Lemma 5.2 there exists a walk with at most $m'/2$ empty triangles which defines an encoding sequence of length $3m'/2$. Since $m' \leq (3/2) \cdot |S_{opt}|$, the encoding sequence is at most 9/4 times optimal.

Finally, suppose that T is not deaf. Except for possibly the start and end vertices of the walk, each ear triangle in any encoding sequence requires at least two empty triangles to exit. By deleting the ears before applying the Christofides heuristic, and inserting the ears into the walk using two empty triangles each, we accommodate these triangles at the same cost as in any optimal solution. Hence these do not contribute to the approximation ratio. $\square$

Acknowledgements

We thank Jarek Rossignac for inspiring this work with his talk at the Third ARO-MSI Stony Brook Workshop on Computational Geometry, and both Claudio Silva and Chris Smith for helpful discussions. We also thank Josh Mittleman for comments and suggestions that improved the paper.

References

[1] D. Avis and H. ElGindy. Triangulating point sets in space. *Discrete Comput. Geom.*, 2:99–111, 1987.

[2] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 23–90. World Scientific, Singapore, 1992.

[3] R. Cassidy, E. Gregg, R. Reeves, and J. Turmelle. IGL: The graphics library for the i860, March 22, 1991.

[4] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. In J. F. Traub, editor, *Sympos. on New Directions and Recent Results in Algorithms and Complexity*, page 441, New York, NY, 1976. Academic Press.

[5] Henry Crapo and Jean-Paul Laumond. Hamiltonian cycles in Delaunay complexes. In *Geometry and Robotics Workshop Proceedings, Toulouse France, 26–28 May 1988*, volume 391 of *Lecture Notes in Computer Science*, pages 292–305, Berlin, 1989. Springer-Verlag.

[6] G. Das, P. Heffernan, and G. Narasimhan. LR-Visibility in polygons. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 303–308, Waterloo, Canada, 1993.

[7] M. B. Dillencourt. A non-Hamiltonian, nondegenerate Delaunay triangulation. *Inform. Process. Lett.*, 25:149–151, 1987.

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

[9] M. B. Dillencourt. Finding Hamiltonian cycles in Delaunay triangulations is NP-complete. In *Proc. 4th Canad. Conf. Comput. Geom.*, pages 223–228, 1992.

[10] M. Garey, D. Johnson, and R. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM J. Computing*, 5:704–714, 1976.

[11] F. Gray. Pulse code communication. United States Patent Number 2,632,058, March 17, 1953.

[12] J. Hershberger. Finding the visibility graph of a simple polygon in time proportional to its size. In *Proc. 3rd Annu. ACM Sympos. Comput. Geom.*, pages 11–20, 1987.

[13] J. O'Rourke, H. Booth, and R. Washington. Connect-the-dots: a new heuristic. *Comput. Vision Graph. Image Process.*, 39:258–266, 1987.

[14] J. O'Rourke and K. J. Supowit. Some $\mathcal{NP}$-hard polygon decomposition problems. *IEEE Trans. Inform. Theory*, IT-29:181–190, 1983.

[15] N. Prabhu. Hamiltonian simple polytopes. Technical Report 90-17, DIMACS, Rutgers Univ., 1990.

[16] C. Savage. Gray code sequences of partitions. *J. Algorithms*, 10:577–595, 1989.

[17] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 37–68. Springer-Verlag, 1993.

[18] Silicon Graphics, Inc. Graphics library programming guide, 1991.

[19] L. H. Tseng and D. T. Lee, "Two-Guard Walkability of Simple Polygons", manuscript, 1992.

[20] H. Wilf. *Combinatorial Algorithms: An Update*. Society for Industrial and Applied Mathematics, Philadelphia, 1989.

[21] M. Yvinec. Triangulation in 2D and 3D space. In *Geometry and Robotics Workshop Proceedings, Toulouse France, 26–28 May 1988*, volume 391 of *Lecture Notes in Computer Science*, pages 275–291, Berlin, 1989. Springer-Verlag.