Recovery of Convex Hulls from External Visibility Graphs

Hazel Everett*
Anna Lubiw†
Joseph O'Rourke‡

Abstract

Although the internal visibility graph does not determine the convex hull of a polygon, it is shown that the external visibility graph determines the hull uniquely, with one exception. An $O(n^2 \log n)$ algorithm is presented for recovering the hull from the external visibility graph when the hull has four or more vertices, the non-exceptional case. The algorithm also identifies the exceptional case: when the hull is a triangle; then which three vertices comprise the hull is underdetermined.

Further, it is shown that the internal and external visibility graphs together still do not uniquely determine triangular hulls.

1 Introduction

The problem of finding a polygon that realizes a given visibility graph seems to be difficult [O'R87]. Progress has only been made by either restricting attention to a subclass of polygons or a subclass of graphs, or by starting with more information than just the visibility graph [O'R93]. In this paper we explore what can be reconstructed from the external visibility graph (definition below). We show that the convex hull is uniquely determined by the external visibility graph, except for one class of polygons.

2 Problem Definition

All polygons considered in this paper are simple polygons, which partition the plane into a bounded interior, the polygon boundary consisting of n vertices and edges, and the unbounded exterior. We assume throughout that the vertices are in general position: no three are collinear. The *internal vertex visibility graph* $G_I(P)$ for a polygon P is an undirected,

*University of Quebec at Montreal, Department of Mathematics and Computer Science, Quebec, C.P. 8888, Montreal, Canada H3C 3P8. everett@mipsmath.math.uqam.ca.

†Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, N2L 3G1. alubiw@waterloo.edu. Supported by NSERC.

‡Department of Computer Science, Smith College, Northampton, MA 01063, USA. orourke@sophia.smith.edu. Supported by NSF grant CCR-9122169.

labeled graph, with a node for each vertex of P, and an arc iff the corresponding vertices are *internally visible* to one another, in the sense that the line segment connecting the vertices is nowhere exterior to P. Note that every edge of the polygon corresponds to an arc in $G_I(P)$. The nodes are labeled with indices $0, 1, 2, ..., n-1$ such that the corresponding vertices form a counterclockwise (ccw) traversal of P's boundary. The *external vertex visibility graph* $G_E(P)$ is defined in the same manner, except that two vertices must be *externally visible* to correspond to an arc: the line segment connecting them is nowhere interior to P. Again every edge of P corresponds to an arc in $G_E(P)$, and the nodes are labeled as in $G_I(P)$. Note we are using improper visibility, in that grazing contact of a line of sight with the polygon's boundary does not block vision. Proper visibility can be quite different, but with our assumption of general position, the only difference between proper and improper visibility is whether the polygon edges are arcs of the graph.

We will abbreviate $G_I(P)$ and $G_E(P)$ to G_I and G_E when the polygon P is clear from the context.

2.1 History

The original *reconstruction problem* posed by Avis and ElGindy is: given G_I, construct a P that realizes it. Note that because the graph is labeled, the Hamiltonian cycle representing the boundary of P is known.

The questions explored in this paper are:

> Given G_I, or G_E, or both G_I and G_E, for a polygon P, can the labels of the vertices of the convex hull of P, $\mathcal{H}(P)$, be uniquely identified?

We will first establish some notation before summarizing our answers.

2.2 Notation

$\mathcal{H}(P)$ (or just $\mathcal{H}$) is the labeled cycle representing the convex hull of P. Throughout we assume that the input graphs are visibility graphs of some (usually unknown) polygon. The problem of *recognizing* visibility graphs is not directly considered. We reserve the term "arc" for a graph edge (a, b), and use "edge" and "diagonal" for the polygon edge or diagonal ab. We will use the symbol "$\Rightarrow$" as shorthand for "uniquely determines," and "$\not\Rightarrow$" for "does not uniquely determine." Finally, we let n be the number of vertices of P and h the number of vertices on $\mathcal{H}(P)$.

2.3 Summary of Results

Here are our answers to the above questions. In the cases where there is not unique determination of the hull, the number of different hulls compatible with the input is listed.

1. $G_I \not\Rightarrow \mathcal{H}$: $\Theta(2^n)$ possibilities.

2. $G_E \Rightarrow \mathcal{H}$ if $h > 3$.

3. $G_E \not\Rightarrow \mathcal{H}$ if $h = 3$: $\Omega(n)$ possibilities.

4. $G_I + G_E \not\Rightarrow \mathcal{H}$ if $h = 3$: $\Omega(n)$ possibilities.

Throughout, our emphasis will be on the ideas and not on algorithm efficiency. Complete proofs may be found in the full version of this paper [ELO93].

3 $G_I \not\Rightarrow \mathcal{H}$

See [ELO93].

4 $G_E \Rightarrow \mathcal{H}$ if $h > 3$

This section presents the main result of this paper: G_E, the external visibility graph, uniquely determines $\mathcal{H}$ for polygons with four or more vertices on their hulls.

4.1 Sketch of Ideas

The main idea behind the hull-identification algorithm is to first use G_E to construct a triangulation of the exterior of P inside $\mathcal{H}$, and then identify hull edges as those edges that do not support a triangle to one side (the exterior). The reason this approach fails when $h = 3$ is that then a hull edge can support triangles to both sides: one in a "pocket," and the other the hull itself. We now make these ideas more precise.

4.2 Definition of External Triangulation

A *pocket* of P is a polygon formed from a nonpolygon hull edge (the pocket *lid*) and the chain of P connecting the endpoints inside the hull. If ab is a nonpolygon hull edge, then the corresponding pocket is the polygon formed of the vertices $(a, b, b-1, b-2, ..., a+2, a+1, a)$. An *external triangulation* of P is a (labeled) graph defined as the union of triangulations of all the pockets, together with the hull edges that are also polygon edges.

4.3 Constructing a Triangulation

A key point is that knowledge of the Hamiltonian circuit representing the boundary of P permits us to infer which arcs of G_E must map to crossing diagonals in any geometric realization. A *geometric realization* (or just *realization*) of G_E is simply a polygon P whose external visibility graph is G_E. Let $C[s,t]$ represent the closed polygonal chain of the boundary of P from s ccw to t, and let $C(s,t)$ be same chain excluding s and t.

Lemma 4.1 *Let (a,b) and (c,d) be two arcs of G_E, and assume without loss of generality that the label indices satisfy $a < b$, $c < d$, and $a < c$. Then the diagonals ab and cd cross in a geometric realization of G_E iff $a < c < b < d$.*

The consequence of this lemma is that we can infer geometric crossing of diagonals in any realization from *topological crossing*, which is captured in the lemma by the condition on the index labels. This permits us to talk unambiguously of arcs of G_E to be *crossing*.

Lemma 4.2 *A maximal set of non-crossing arcs in $G_E(P)$ corresponds to an external triangulation of P.*

We will defer discussion of an algorithm for constructing a triangulation to Section 6.

4.4 Identifying Hull Edges

The following lemma is the key to identifying which arcs of G_E must be hull edges. Say an arc $e = (a, b)$ *supports a triangle* in G if both endpoints of e are adjacent to some vertex c. Say that e supports triangles *to both sides* if both endpoints of e are adjacent to c and to d, and (a, b) and (c, d) are crossing diagonals.

Lemma 4.3 *For a polygon with $h > 3$,*

1. *polygon arc $(a, a + 1) \in G_E$ is a hull edge in any realization iff it does not support a triangle in G_E.*
2. *nonpolygon arc $(a, b) \in G_E$ is a hull edge in any realization iff it does not support a triangle in G_E to one side.*

5 Polygons with triangular hulls

Lemma 4.3 provides an easy method for identifying arcs of $G_E(P)$ that must be hull edges in any realization if $h > 3$. But if $h = 3$, i.e., if the hull of P is a triangle, then in fact it is not always possible to identify a hull edge from G_E. We will illustrate with an example this later.

5.1 Distinguishing between $h = 3$ and $h > 3$

Lemma 5.1 *For a polygon P with $h = 3$,*

1. *if polygon edge $(a, a+1) \in G_E$ is a hull edge of P, then it supports one triangle in G_E.*
2. *if nonpolygon edge $(a, b) \in G_E$ is a hull edge, then it supports triangles to both sides in G_E.*

Lemmas 4.3 and 5.1 together yield a method for determining from $G_E(P)$, whether $h = 3$, as follows. If every polygon arc supports a triangle in G_E, and every nonpolygon arc supports a triangle to either side, then it must be that $h = 3$. For if $h > 3$, a hull edge that did not support a triangle to one side would have been found, since Lemma 4.3 is a necessary and sufficient characterization. On the other hand, if there is an edge in G_E that does not support a triangle to one side, Lemma 5.1 shows that $h \neq 3$, and therefore $h > 3$.

5.2 $G_E \not\Rightarrow \mathcal{H}$ if $h = 3$

See [ELO93].

6 Hull Algorithm

The algorithm for reconstructing the convex hull from the external visibility graph will now be sketched. The algorithm has four main steps, as follows.

Algorithm Hull: $G_E \Rightarrow \mathcal{H}$ IF $h > 3$	
1. Find an external triangulation T of P from G_E.	$O(n^2 \log n)$
2. Mark edges of T *supporting* or not.	$O(n)$
3. if every edge is marked *supporting* then $h = 3$	$O(n)$
4. else output as hull all unmarked edges	$O(n)$

We only discuss Step 1 here. A maximal set of noncrossing arcs can be found by selecting an arc, discarding all arcs that cross it, and repeating this process until all arcs have been selected or discarded.

```
Step 1: TRIANGULATION
G ← G_E
while not all arcs of G selected
        Select arc (a, b) ∈ G.
        for each vertex c
                Delete from G all arcs (c, d) that cross (a, b).
Output T = G.
```

We summarize in a theorem.

Theorem 6.1 *Algorithm Hull, in $O(n^2 \log n)$ time, identifies the unique convex hull of P from $G_E(P)$ in the case that P has more than 3 hull vertices, or detects that P has a triangular hull without identifying its vertices.*

7 $G_I + G_E \not\Rightarrow \mathcal{H}$ if $h = 3$

It is tempting to think that the addition of the internal visibility graph G_I to the input information might disambiguate the many possible hull choices left by knowing only G_E when $h = 3$. Unfortunately G_I still leaves the hull underdetermined. Isolated examples of this phenomenon were presented in [O'R90]. Here we present an example in Fig. 1 demonstrating that there may be as many as $\Omega(n)$ polygons with distinct triangular hulls simultaneously realizing a given G_E and G_I.

8 Discussion

The problem of finding a polygon P that realizes G_I seems to be quite difficult. Since it seems hard, two approaches have beee tried: restrict aspects of the problem until it becomes tractable, or augment the input. Our contribution can be viewed as an instance of the latter strategy: we have shown that the external visibility graph permits reconstructing the convex hull of the polygon in most cases.

We close with three more specific open problems:

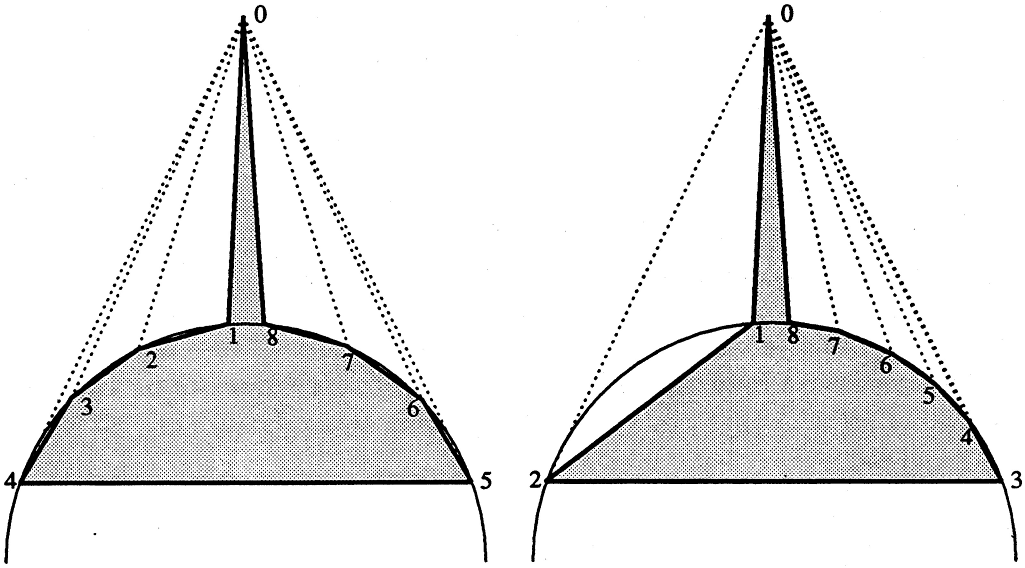

Figure 1: Polygons with identical G_E and G_I. (a) $\mathcal{H}$ = (0,4,5). (b) $\mathcal{H}$ = (0,2,3).

1. Find a polygon that realizes given internal and external visibility graphs – an "easier" reconstruction problem than Avis and ElGindy's, but apparently still hard.

2. Characterize those polygons P for which $G_I(P) = G_E(P)$. That this class is nonempty is demonstrated by a triangle. The question may be asked for both labeled and unlabeled graphs.

3. More generally, characterize those polygons P that realize specific pairs of graphs G_1 and G_2: $G_I(P) = G_1$ and $G_E(P) = G_2$. For example, in Fig. 1, two different polygons realizing particular pairs of graphs are shown.

References

[ELO93] H. Everett, A. Lubiw, and J. O'Rourke. Recovery of convex hulls from external visibility graphs. Technical Report 024, Smith College, January 1993.

[O'R87] J. O'Rourke. *Art Gallery Theorems and Algorithms.* Oxford University Press, Oxford, England, 1987.

[O'R90] J. O'Rourke. Recovery of convexity from visibility graphs. Technical Report 90.4.6, Smith College, 1990.

[O'R93] J. O'Rourke. Computational geometry column 18. *Internat. J. Comput. Geom. Appl.*, 3(1):107–113, 1993. Also in SIGACT News 24:1(1993) 20–25.