Geometric Algorithms (2IL20)

Dirk Gerrits & Christian Luijten

June 23, 2006

Research Project Report

Dirk Gerrits & Christian Luijten
Geometric Algorithms (2IL20)
Department of Mathematics & Computer Science
Technische Universiteit Eindhoven

1 Introduction

The report at hand describes a solution to the problem of Construction sequences as described in the List of Problems document for the Geometric Algorithms course 2006 at the Technische Universiteit Eindhoven.

We first give a description of the problem as given on the course website. Next, we describe the idea that led to our solution in Section 2.

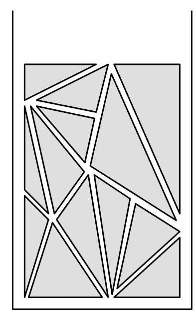
The problem description asks for a proof that a construction sequence always exists, this proof is given in Section 3.

Once we know that a construction sequence exists, we describe an algorithm which constructs such a sequence in Section 4.

To round off the project, we draw conclusions in Section 5.

1.1 Problem description

Imagine that we would like to manufacture a product that consists of several parts. To simplify the problem we will consider only the planar case where all parts are triangles. We assemble the product by vertically translating each part into its proper position. This implies that we first have to move the bottommost parts into position, then the next higher ones, etc. Prove that there is always a construction sequence – if the parts are triangles – and give an $O(n \log n)$ *algorithm that computes such a sequence.*



2 Solution idea

In a construction sequence, a triangle may not be placed before all the triangles below it are placed. This gave us the idea to use topological sorting.

Basically, we go through all the triangles and create a link between adjacent triangles if one is above the other. The link is directed from the top to the bottom triangle. This is illustrated in Figure 1.

With these links, a triangle can be placed safely if all the triangles to which it has outgoing links have been placed already. So when we perform a topological sorting on the graph induced by these links, we get a construction sequence.

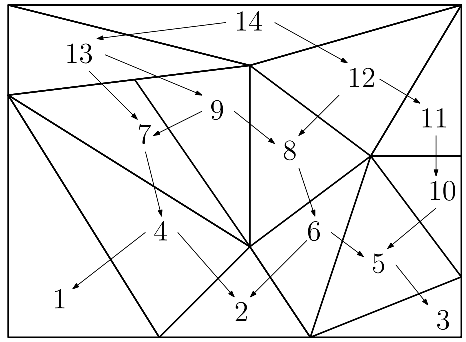


Figure 1: Construction of a topological order

3 Existence proof

To prove that a construction sequence always exists, we need to prove that we can always find this topological order, and with it the corresponding construction sequence. To do this we first need to make some notions more precise.

Definition 1: The *normal* of a triangle edge is a unit vector perpendicular to the edge, pointing out of the triangle. We call a triangle edge a *downward edge* if its normal has a negative y-component, an *upward edge* if its normal has a positive y-component, and a *vertical edge* if its normal has a zero y-component.

Definition 2: Triangle a is *directly below* triangle b, and b is *directly above* a, iff a is incident to the interior of a downward edge of b. For a set $\mathcal{T}$ of triangles we define the *directed graph* $\mathcal{D}(\mathcal{T})$ such that $\mathcal{D}(\mathcal{T})$ has a vertex for every triangle in $\mathcal{T}$, and for all $a, b \in \mathcal{T}$ it has an edge from a to b iff b is directly below a.

Theorem 1: Graph $\mathcal{D}(\mathcal{T})$ is acyclic.

Proof. By contradiction. Suppose we have a cycle of m triangles $t_1, \ldots, t_m, t_1 = t_{m+1}$. For every i, t_i has a downward edge e_i incident to t_{i+1}, so the cycle of m triangles induces a cycle of m downward edges $e_1, \ldots, e_m, e_1 = e_{m+1}$.

Going from one edge to the next, the normal makes a turn, either to the left or to the right. The difference between the sum of the angles of the left turns, and the sum of the angles of the right turns, is 360°. So together, all these turns sweep out a 360° arc. (See Figure 2.) Of that arc, one 180° subarc corresponds to upward edges (positive y-component), so none of the turns can end there.

This means that at least one turn must be greater than 180°, but all the angles in a triangle are smaller than 180°, so none of our turns can be greater than 180°, and we have a contradiction. Hence $\mathcal{D}(\mathcal{T})$ cannot have cycles. □

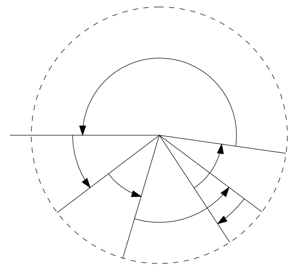


Figure 2: The turns of the edge normals sweep out a 360° arc

Because $\mathcal{D}(\mathcal{T})$ is acyclic, it has a (not necessarily unique) topological order, and therefore a construction sequence always exists.

4 Construction algorithm

We assume we are given the triangles $\mathcal{T}$ in a doubly-connected edge list structure. Because an edge in our input can be incident to multiple edges of other triangles, the structure is different from the standard DCEL by having a list of twin pointers for each edge, rather than a single twin pointer.

With this data structure it's a simple matter to construct the graph $\mathcal{D}(\mathcal{T})$, and then we can simply feed that graph to a topological sorting routine:

Algorithm

```
1: initialize D(T) with no edges and a vertex for every triangle
2: for every half-edge e in the DCEL do
3:   if e is a downward edge then
4:     for every twin e' of e do
5:       add an edge to D(T) from IncidentFace(e) to IncidentFace(e')
6: topologically sort D(T) and return the produced order as the construction
   sequence
```

Lines 2-3 loop through all downward half-edges, of which there are at most $2n$. Over all iterations of lines 2-3, line 5 is executed exactly as many times as $\mathcal{D}(\mathcal{T})$ will have edges. Because $\mathcal{D}(\mathcal{T})$ is a planar graph, it has at most $3n-6$ edges, according to Euler's formula. Thus the construction of $\mathcal{D}(\mathcal{T})$ takes $\Theta(n)$ time.

Line 6, i.e. topological sorting, can be implented to run in $\Theta(|V|+|E|)$ time where $|V|$ is the number of vertices and $|E|$ is the number of edges. (See for example the algorithm in [CLRS, Chapter 22], which works even if $\mathcal{D}(\mathcal{T})$ is not fully connected.) Since we already argued that $|V|=n$, $|E|=O(n)$, the topological sorting step also takes $\Theta(n)$ time.

5 Conclusion

The assignment asked for a $O(n \log n)$ algorithm, which could have been achieved with a plane sweeping approach as discussed in the course lectures. We have taken an alternative, graph-theoretic approach, based on topological sorting. We have proven that (for triangles) such a topological order always exists, and gave a $\Theta(n)$ algorithm to compute it. This order is exactly a valid construction sequence.

References

[CLRS] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. Introduction to Algorithms (2nd edition). MIT Press / McGraw-Hill, 2001.