Application of fractional casacading to computational geometry

Vishwali Mhasawade
vishwalim@nyu.edu

Abstract

This document aims to illustrate the application of fractional cascading to determine the edges of a polygonal path, P that intersect with a given line l first introduced by Chazelle and Guibas [1986]. This can be done in $\mathcal{O}(\log n + k)$ steps where n is the number of vertices on the polygonal path and k represents the number of edges of the polygonal path p that intersect with the line l.

1 Goal

Given the coordinates of vertices on the polygonal path $\{(x_i, y_i) \mid i = 1, \ldots n\}$ and line, l represented by the equation $y = mx + c$ where m represents the slope with the $+x$ axis and c represents the intercept on the y axis; determine the edges of the polygonal path that intersect with the given line l.

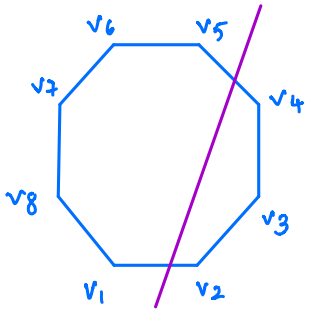

Figure 1: Setting: An octagon with the query line.

2 Illustration

For simplicity we first consider a simple polygon (i.e. a polygon that does not have any crossings or holes). We begin by considering an octagon (polygon that has eight equal length sides) and a line. This is represented in Fig 1.

Trivially this can be done in $\mathcal{O}(n)$ by checking if the line intersects with each of the polygonal edge. While this seems reasonably if the number of edges in the polygonal path are small but becomes expensive with increasing polygonal path edges. Fractional cascading can help in improving this to $\mathcal{O}(\log n + k)$. This is especially crucial when only a small number of polygonal edges intersect with the line even if the total number of edges is high. The method provides a series of preprocessing steps that can reduce the query time for intersection of edge and line considerably which are listed below.

Step 1. Split the given polygonal path P into two subparts of $n/2$ and $n/2$ vertices respectively. Let us call these subparts P_{left} and P_{right}. The split is performed by considering the median edge. Each group contains edges in the order of non-decreasing slopes[1]. Here, P_{right} consists of vertices $\{V_2, V_3, V_4, V_5\}$ and the respective edges originating at these vertices. P_{left} consists of vertices $\{V_6, V_7, V_8, V_1\}$ and the egdes originating at them. Figure 2 indicates the vertices in the subparts after splitting along with the slopes of edges originating at the specific vertices.

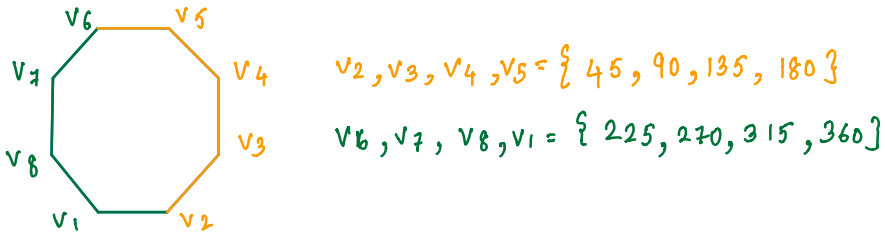

Figure 2: Vertices in P_{left} and P_{right} with the slopes of the edges originating at the specific vertices.

Step 2. For a specific path P^* we construct a hierarchical representation of the convex hulls of vertices V^* in P^* as follows:

(a) We recursively split the edges we have (say j) into two parts consisting of $j/2$ and $j/2$ respectively (if j is even else $j/2 - 1$ and $j/2 + 1$) while maintaining the order of the vertices (in regard to non-decreasing slopes) unless we are left with one edge in each of the group. The leaves in the tree consist of the actual edges from the polygonal path P while the intermediate nodes consist of groups of edges in the same order as in the original polygonal path P. This is represented in Figure 3.

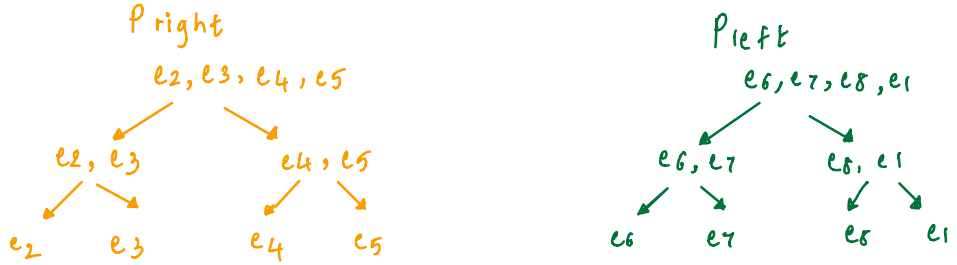

Figure 3: Hierarchical structure originated by splitting the edges recursively in each subpart.

(b) Next we begin by constructing convex hulls at leaf level by considering the edges. The left and right convex hulls can be combined in linear time using standard algorithms like rotating calipers [Toussaint, 2014]. The convex hulls at each level in the hierarchy are represented in Figure 4 for both left and right subtrees.

(c) Corresponding to the convex hulls at each of the nodes in the tree we also store the edges in present on the hull. This may introduce some new edges that may not be present in the origial polygonal path P as presented in Figure 5.

(d) Once we have the edges in each of the convex hulls in a hierarchical manner, we perform fractional cascading. This is different than fractional cascading in a linear chain. The difference is illustrated in Figure 6. For our specific example this is described as follows. Since the degree in the hierarchical structure is two (each parent has two children), the fraction of augmentation should be $< 1/2$. Here, we consider it to be $1/3$. We represent

[1]Slope is the angle made by the edge with the positive x axis.

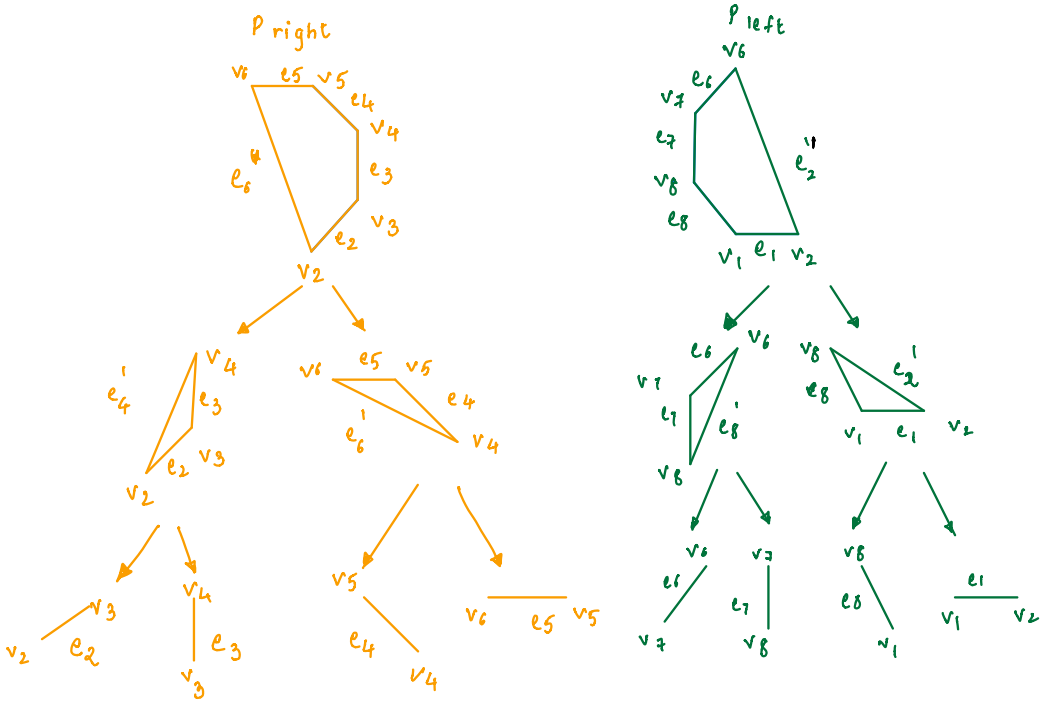

Figure 4: Convex hulls of vertices at each hierarchical level. The left and right convex hulls can be combined in $\mathcal{O}(n)$ time using standard algorithms like rotating calipers.

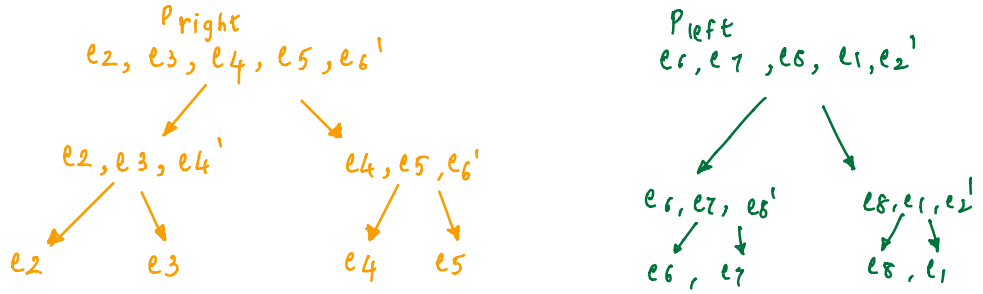

Figure 5: Edges in the convex hull at each hierarchical level. Some of these edges may not be present in the original polygonal path. For example, e'_4, e'_6 are the new edges at level 2 in right subtree originating due to the convex hull at that level while e'_8, e'_2 are the new edges in left subtree.

the elements propagated by the left child by x and those propagated by the right child by x. This is shown in Figure 7.

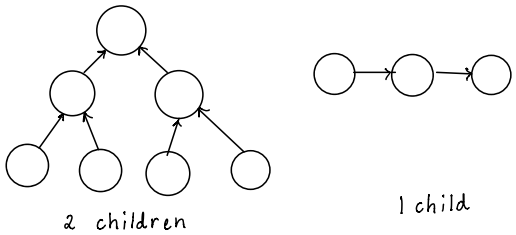

Figure 6: Fractional cascading with more than one child node.

(e) We next create bridges to ensure quick access between lists. Since we have elements propagated by two lists we need to have two pointers from original elements, one to the left child element and other to right child element. We create a bridge to the largest promoted

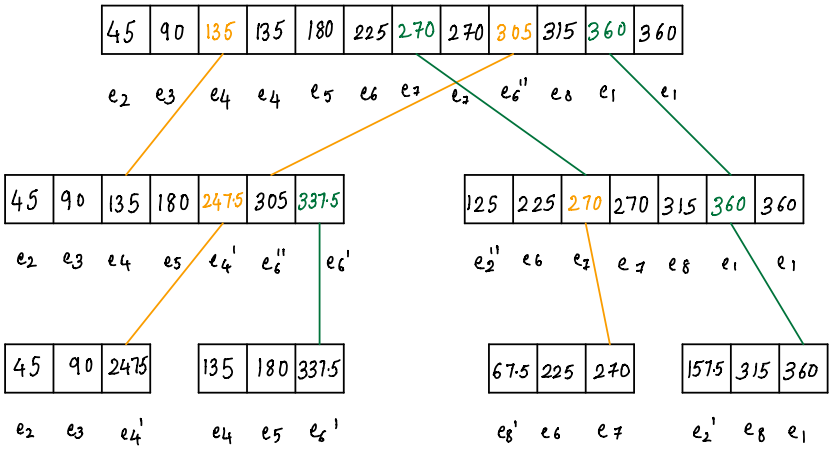

Figure 7: Fractional cascading structure with elements propagated from two children to the parent node.

element from both the children. However, if there is no promoted element greater than the original element, we link to the last promoted element (by index) from the child list. We also create a bridge from the promoted to the next larger original element. For our specific example of the cascading structure presented in Figure 7, the bridges are represented in Figure 8.

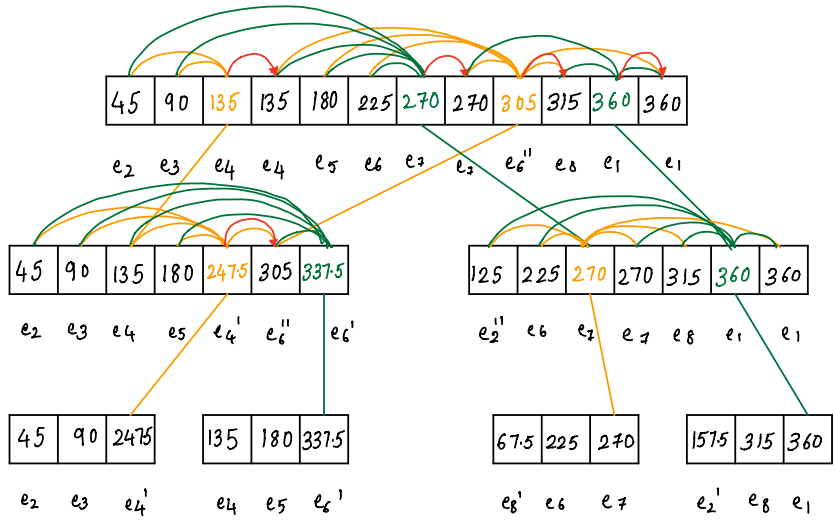

Figure 8: Bridges from the original to the two promoted elements are represented by $\rightarrow$ and $\rightarrow$ whereas bridges from promoted to original are represented by $\rightarrow$.

(f) **A line intersects a polygonal path only if it intersects with the convex hull of the polygonal path**.

- To check if a line intersects a convex hull we find vertices of the convex hull where tangents parallel to the line can be drawn (tangents with slope m and $m + 180$)
- Say these tangents have equations: $t_1 : y = mx + c_1$ and $t_2 : y = (180 + m)x + c_2$. We then determine if our line $y = mx+_c$ lies in between t_1 and t_2. If yes then the line intersects with the convex hull else not.

(g) We need to find vertices in each of the convex hulls (original and intermediate nodes in the hierarchical structure) where tangents can be drawn. We then check if the line intersects

with the convex hull $Ch(P^*)$. If it does not then we do not need to check the intersection of the children of $Ch(P^*)$ reducing the computations.

(h) Fractional cascading enables us to find a vertex (with slope m) in $\mathcal{O}(\log n)$ by performing binary search on the original convex hull. We can then locate the vertices in each list in constant $\mathcal{O}(1)$ with the help of bridges.
Let us see how this works for a line with slope $m = 60$ (and antislope $m = 240$). To do this we inspect the slope x_i just greater than m such that the previous element $x_{i-1} < m$.
After finding the element in list we follow the pointers propagated from left and right child. Moving to the child list we only need to look at the positions $i - 1$ and $i - 2$ to find the element in that list. Figure 9 shows how the slopes 60 and 240 can be found in $\mathcal{O}(\log n)$ steps at the topmost level and in $\mathcal{O}(1)$ steps at each of the lower levels.

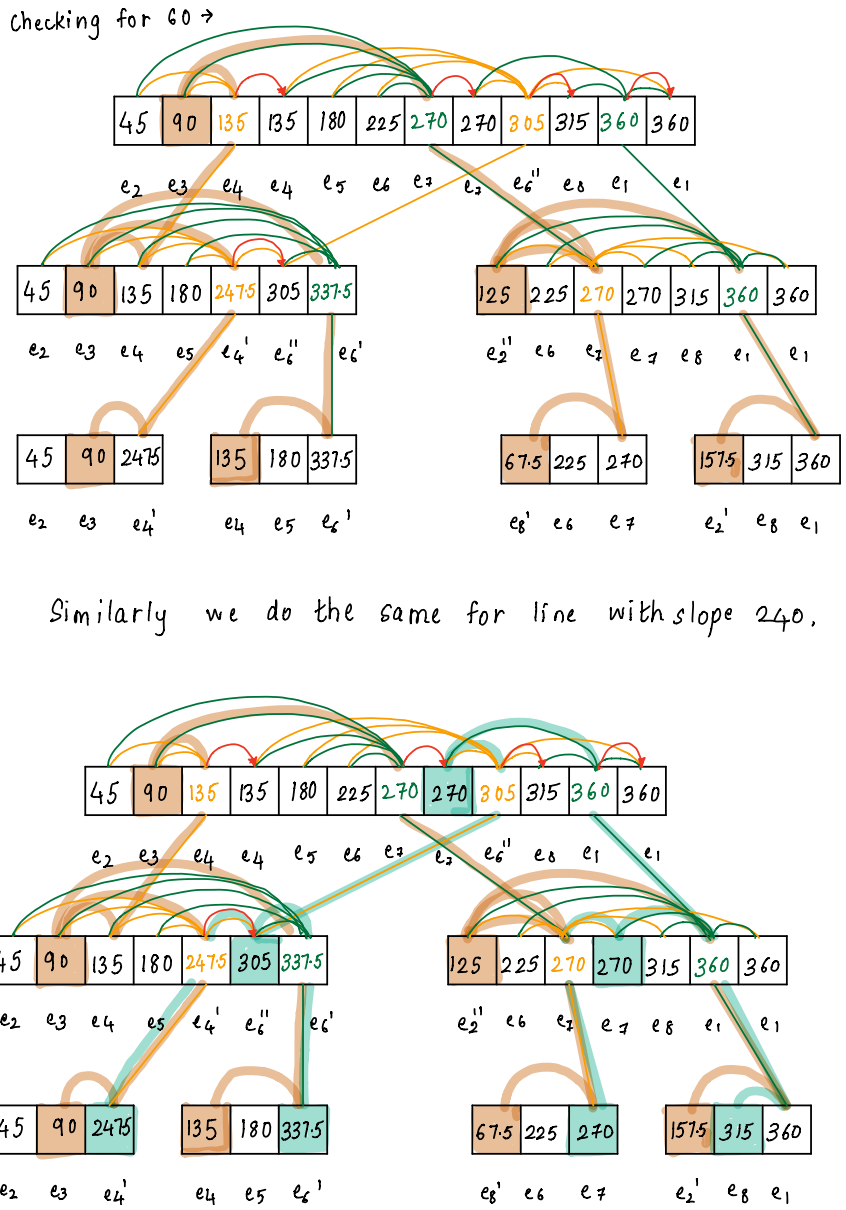

Figure 9: Finding vertices in each of the convex hull where tangents parallel to $m, (180 + m)$ can be drawn. This can be done in $\mathcal{O}(\log n)$ steps since we only need to binary search at the topmost level and then follow the pointers to get the positions in child nodes in $\mathcal{O}(1)$ time.

(i) After locating the vertices in all the lists we search the convex hull at the top to see if the line intersects. If it does then we begin processing the left and right child for the same. The vertices in convex hull at each hierarchical level where tangents parallel to query line can be drawn are represented in Figure 10.

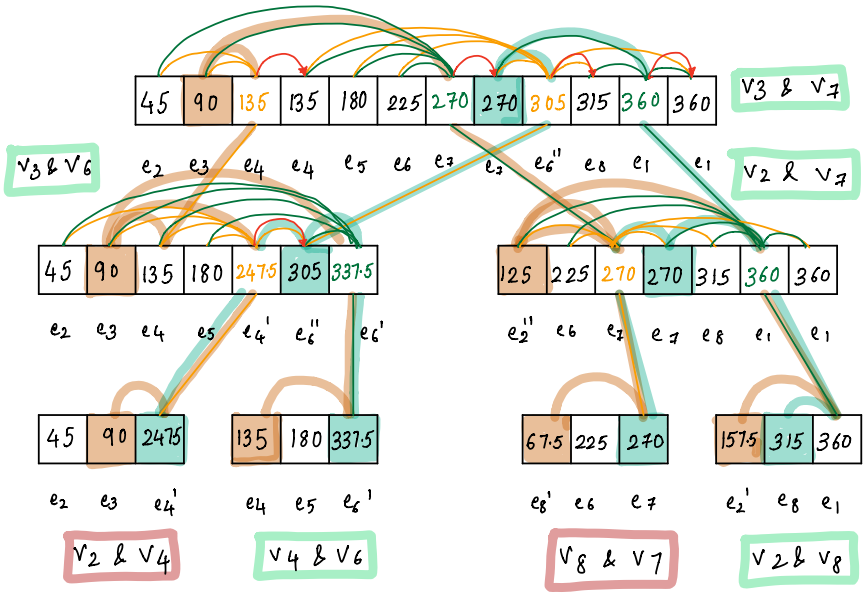

Figure 10: Vertices in each convex hull where tangents parallel to $m = 60, 240$ can be drawn.

(j) At the last step we only check the edges in the lowest convex hull for intersection. In this case the result is edges e_1 and e_4.

3 Conclusion

We have illustrated how fractional cascading can be used to speed up the process of determining the intersecting edges of a polygonal path with a line. Although here it is described for a simple polygon (octagon) the same process can be applied to any polygonal path while ensuring the fractional of augmentation is suitable.

Acknowledgements

I would like to thank Professor Greg Aloupis for clarifying the concepts and providing pivotal feedback.

References

B. Chazelle and L. J. Guibas. Fractional cascading: Ii. applications. *Algorithmica*, 1(1-4):163–191, 1986.

G. T. Toussaint. The rotating calipers: An efficient, multipurpose, computational tool. In *The International Conference on Computing Technology and Information Management (ICCTIM)*, page 215. Society of Digital Information and Wireless Communication, 2014.