Enumerating Trichromatic Triangles Containing the Origin in Linear Time

Amr Elmasry and Madhusudan Manjunath
Max-Planck-Institut für Informatik, Saarbrücken, Germany

Abstract

Consider a set P of n points in the plane each associated with one of a constant number of colours. We give an output-sensitive algorithm that runs in $O(n + |T|)$ time to enumerate the set of triangles T, such that each triangle contains the origin, its vertices are in P and have three distinct colours.

1 Basic framework

We assume, without loss of generality, that the point set lies on the unit circle. The uncoloured version of the problem has been studied previously, see [1, 2]. We use the following observation [1]:

Lemma 1. *Given a triangle with vertices on the unit circle, the triangle contains the origin if and only if for any of its vertices v, the point $-v$ lies on the smaller arc defined by the other two vertices.*

With Lemma 1 in mind, we project the points that are on the upper half of the unit circle onto the lower half using the mapping $x \rightarrow -x$. Let P_1 be the set of points originally on the lower half, and P_2' be the points originally on the upper that were projected onto the lower half of the unit circle. For a point p, we denote the angle it forms with the x-axis by $\alpha(p)$. Using Lemma 1, we can reduce our problem to the problem of enumerating certain subsequences $\langle s_1, s_2, s_3 \rangle$ of points in $P_1 \cup P_2'$, such that: s_1, s_2 and s_3 all have distinct colours, $\alpha(s_1) \leq \alpha(s_2) \leq \alpha(s_3)$, and either $s_1, s_3 \in P_1$ and $s_2 \in P_2'$ or $s_1, s_3 \in P_2'$ and $s_2 \in P_1$.

As a first attempt, we sketch a sorting-based algorithm to enumerate the triangles in T. We first sort the points in $P_1 \cup P_2'$ by the angle they form with respect to the x-axis. We then construct lists of consecutive points that have the same colour and orientation. Using this information, we can enumerate the set of triangles T in $O(|T|)$ time. Still, this algorithm has time complexity $O(n \log n + |T|)$.

If the size of the set T is $o(n \log n)$, the question of whether we can do better than $O(n \log n)$ arises. Essentially, this is equivalent to asking whether we can "avoid" sorting. To answer this question, we consider the approach for the uncoloured case followed in [1]. The basic idea of this approach is to efficiently partition the input sequence of points into consecutive subsequences or blocks. A block is defined in [1] as the maximal set of consecutive points (if they are sorted by angle) contained in either of the sets P_1 or P_2'. The points within a block are not sorted though. The intuition behind this partitioning is that if the number of blocks is "large", then $|T|$ is also "large". Accordingly, the running time of the algorithm, which depends on the number of blocks, will be dominated by $|T|$.

In the coloured version, a similar notion of a block cannot be used. Consider the case where all the points have the same colour. In such a case, the number of blocks can be large while T is empty. In order to make the running time depend on $|T|$, the partitioning must accordingly be "colour-sensitive".

2 The algorithm

Each point in the set $P_1 \cup P_2'$ has the following two attributes associated with it: its orientation (whether it belongs to P_1 or P_2') and its colour. We say that a point is of the type t_i if it belongs to P_1 and its colour is i, and that it is of the type t_i' if it belongs to P_2' and its colour is i.

Recall that a triangle in T is represented by a sequence of points $\langle s_1, s_2, s_3 \rangle$. We say that a triangle has the configuration (t_i, t_j', t_k) if s_1, s_2 and s_3 are points of the types t_i, t_j' and t_k respectively, and that it has the configuration (t_i', t_j, t_k') if s_1, s_2 and s_3 are points of the types t_i', t_j and t_k' respectively.

Since there is a constant number of such configurations, we fix a configuration (e.g. (t_0, t_1', t_2)) and describe an algorithm to enumerate the triangles of this configuration. The algorithm must then be applied for all configurations. Next, we describe the key steps of the algorithm.

i. Partitioning the point set into blocks: Construct a list L having only points of the types t_0, t'_1 and t_2. Compute the minimum angle α_0 formed with the x-axis by a point of type t_0, and the maximum angle α_2 formed with the x-axis by a point of type t_2. Delete any point p in L that satisfies $\alpha(p) \leq \alpha_0$ or $\alpha(p) \geq \alpha_2$ (such points cannot be vertices of a triangle with the configuration (t_0, t'_1, t_2)).

We first identify the boundaries of the blocks as follows. A maximal subsequence $\langle b_0, b_1, \ldots, b_{s-1} \rangle$ of points in the list L is identified such that: b_0 is a point of type t_0, b_1 is a point of type t'_1, b_2 is a point of type t_2, b_3 is a point of type t_0, and so on up to b_{s-1} that will be of type t_2. In other words, the points of $\langle b_0, b_1, \ldots, b_{s-1} \rangle$ are of the types t_0, t'_1 and t_2 in cyclic order, and s is a multiple of 3. Furthermore, these points are in sorted order with respect to the angles they form with the x-axis.

We then construct each block B_i, for all $1 \leq i \leq s-1$, by finding all the points in L such that the angles they form with the x-axis lie in the interval $[\alpha(b_{i-1}), \alpha(b_i)]$. Finally, we construct the last block B_s from the remaining points in L excluding those points of type t_0 (such points will not be vertices of any valid triangle, otherwise s is not maximal).

ii. Enumeration of interblock triangles: An interblock triangle is a triangle with no two of its vertices in the interior of the same block. Let $L'(B, t)$ be the list of points of type t that are contained in block B. For each block B_j, construct lists $L'(B_j, t_0)$, $L'(B_j, t'_1)$ and $L'(B_j, t_2)$. This is done in a straightforward manner by traversing the list of points associated with the block. Repeat the following operations for all j from 2 to $s-1$. Incrementally construct the list $\mathcal{L}_0(j)$ as the concatenation of the lists $L'(B_i, t_0)$ for all $i < j$, and the list $\mathcal{L}_2(j)$ as the concatenation of the lists $L'(B_i, t_2)$ for all $i > j$. For every point v'_1 in $L'(B_j, t'_1)$, enumerate all the triangles with vertices (v_0, v'_1, v_2) such that v_0 is a point in $\mathcal{L}_0(j)$ and v_2 is a point in $\mathcal{L}_2(j)$. Note that the list $\mathcal{L}_0(j+1)$ can be obtained by appending $L'(B_j, t_0)$ to $\mathcal{L}_0(j)$, and the list $\mathcal{L}_2(j+1)$ can be obtained by deleting $L'(B_j, t_2)$ from $\mathcal{L}_2(j)$.

iii. Partitioning blocks into subblocks: An intrablock triangle is a triangle with two vertices in the interior of the same block. Note that there are no triangles with three vertices in the interior of the same block. In order to enumerate the intrablock triangles, we need further information about the ordering of the points within each block. A natural approach is to further divide a block into subblocks.

Observe that there cannot be a triangle (with the configuration (t_0, t'_1, t_2)) with two of its vertices in the interior of a block that ends with a point of type t'_1. Accordingly, we need not subdivide such blocks. For every block that ends with either a type t_0 or t_2 point, we further divide it into subblocks. Each subblock is a maximal consecutive sequence (if ordered by the angle formed with the x-axis) of points of the same type. But, the construction of subblocks must be done in a careful order. Let $H_1, H_2, \ldots, H_l$ be the subblocks of a given block in their angle order. The partitioning procedure should identify the sublocks in the following alternating manner: $H_1, H_l, H_2, H_{l-1}, H_3, \ldots, H_{\lfloor l/2 \rfloor + 1}$.

iv. Enumeration of intrablock triangles: Note that the interior of each block contains points of at most two types. Furthermore, each subblock contains points of the same type. In order to enumerate the intrablock triangles, we consider subblocks containing points of the type t'_1. Let U_j be such a subblock. For every point v'_1 of the type t'_1 in U_j, we enumerate triangles of the form (v_0, v'_1, v_2) where v_0 and v_2 are points in the subblocks U_i of type t_0 and U_k of type t_2 respectively, where $i < j < k$. In addition, we make sure that either both U_i and U_j or both U_j and U_k are from the same block. The details of the enumeration procedure are similar to that of the interblock triangles.

Next, we state the key lemmas which directly imply the linear-time complexity of our algorithm.

Let $\tilde{n}$ be the total number of points in all blocks.

Lemma 2. *Enumerating the interblock triangles with the configuration (t_0, t'_1, t_2) requires $O(\tilde{n} \cdot s + n + |T_1|)$ time, where T_1 is the set of such triangles.*

Lemma 3. *For $s \geq 6$, the total number of interblock triangles with the configurations (t_0, t'_1, t_2) or (t_2, t'_1, t_0) is $\Omega(\tilde{n} \cdot s)$.*

Let B be a given block that does not end with a point of the type t'_1. Let $H_1, H_2, \ldots, H_l$ be the subblocks of B in their angle order, and $\tilde{m}_i$ be the number of points in H_i. Let T_2 be the set of intrablock triangles, with the configuration (t_0, t'_1, t_2), each having two of its vertices in the interior of B.

Lemma 4. *Enumerating the triangles in T_2 requires $O(\sum_{i=1}^{l} \tilde{m}_i \cdot \min\{i, l-i+1\} + |T_2|)$ time.*

Lemma 5. $|T_2| = \Omega(\sum_{i=1}^{l} \tilde{m}_i \cdot \min\{i-1, l-i\})$.

References

[1] A. Elmasry and K. Elbassioni, Output-sensitive algorithms for enumerating and counting simplices containing a given point in the plane, CCCG 2005, 248–251.

[2] A. Elmasry and K. Makino, Finding intersections of bichromatic segments defined by points, CCCG 2007, 269–272.