Finding Specified Sections of Arrangements: 2D Results

P. Bose[1] F. Hurtado[2] H. Meijer[3] S. Ramaswami[4] D. Rappaport[3]
V. Sacristán[2] T. Shermer[5] G. Toussaint[6]

Abstract

Given a configuration $\mathcal{C}$ of geometric objects in R^2 (called the *input configuration*), a *target configuration* $\mathcal{T}$ of geometric objects in R^1, and a class $\mathcal{S}$ of allowable sectioning lines we consider in this paper several variations on the following problem: "Is there a line $S \in \mathcal{S}$ such that the section $S \cap \mathcal{C}$ is equivalent by rigid motion to the target $\mathcal{T}$?"

1 Introduction

Mathematical tomography deals with a set of "techniques of reconstructing internal structures in a body from data collected by detectors (sensitive to some sort of energy) outside the body" [3].

When density functions are replaced by geometric objects (for example we have a convex polytope instead of a body with non-constant internal density) then geometric information – shape, measure, ... – is the detected data, and we arrive to the area of *Geometric Tomography*, a topic that has recently emerged as a well defined domain[2] of research.

Computational Geometric Tomography is a natural name for an area grouping results in which the emphasis is on the algorithmic aspect of the problems above. It is illuminating to first distinguish between *direct* and *inverse* problems. In direct problems the input is a geometric object and the scope is to efficiently compute a section or a projection with prescribed properties. In inverse problems, which are at the core of mathematical tomography, sections and/or projections are the input and the aim is to determine, verify, reconstruct or approximate the object, or some of its properties. Direct problems, besides its intrinsic interest, often provide the basis for solving inverse problems. Both types of problems have attracted a lot of attention in Computational Geometry.

In this paper we study some fundamental two dimensional recognition problems, involving objects and arrangements that are typical in Computational Geometry (3D analogous results are described in a companion paper [1]). Specifically, given a configuration $\mathcal{C}$ of geometric objects in R^2 (called the *input configuration*), a *target configuration* $\mathcal{T}$ of geometric objects in R^1, and a class $\mathcal{S}$ of allowable sectioning lines we consider several variations on the following problem: "Is there a line $S \in \mathcal{S}$ such that the section $S \cap \mathcal{C}$ is equivalent by rigid motion (translation plus rotation) to the target $\mathcal{T}$?" In the affirmative, the algorithms will report all the solutions, if there are a finite number. When there is an infinite set of parallel lines that give the same combinatorial solution, only one of them will be reported. This framework is denoted throughout the paper as the *sectioning problem*.

2 Two dimensional problems

The classes of input configurations that we consider are sets of line segments (where the target is a set of points), sets of lines (where the target is a set of points), sets of discs (where the target is a set of intervals), and sets of polygons (where the target is a set of intervals). We consider any line, horizontal lines and lines through the origin, as allowable sectioning classes. We use n to denote the number of objects in $\mathcal{C}$, and k for the number of objects in $\mathcal{T}$.

The following table summarizes our results.

Inputs	sectioning class		
	horizontal	origin	any
line segments	$O(n^2)$	$O(n^2)$	$O(n^3)$
non-par. lines	$\Theta(n \log n)$	$O(n^2)$	$O(n^3)$
polygons	$O(n^2)$	$O(n^2)$	$O(n^3)$
discs	$O(n^2)$	$O(n^2)$	$O(n^3)$

As shown in this table, if $\mathcal{C}$ is a set of non-parallel lines the sectioning problem is solvable in $O(n \log n)$

[1]Carleton University, Ottawa.
[2]Universitat Politècnica de Catalunya, Barcelona. Partially supported by Proyecto DGES-SEUID PB96-0005-C02-02.
[3]Queen's University, Kingston.
[4]Rutgers University, Camden.
[5]Simon Fraser University, Burnaby.
[6]McGill University, Montréal.

time, which matches the lowerbound on the complexity of the problem. If $\mathcal{C}$ is an arbitrary set of lines and $k = n$ we can still solve the sectioning problem in $O(n \log n)$ time. However if $k < n$ we do not know whether an $O(n \log n)$ algorithm exists.

To give a flavour of our results we sketch the proof for the case where the input is a set of non-crossing line segments and the target set is obtained by a horizontal sectioning line. This result can also be applied when the input is a set of non-intersecting polygons.

Theorem 1 *Given an input configuration $\mathcal{C}$, of non-crossing line segments, and a target configuration, $\mathcal{T}$ of points in R^1, we can determine in $O(n^2)$ time whether there is a horizontal sectioning line S, such that $S \cap \mathcal{C}$ is equivalent by rigid motion to $\mathcal{T}$.*

PROOF (Sketch).

Sort the points in $\mathcal{T}$. Let $p_0, p_1, \cdots, p_{k-1}$ denote the sorted points. Compute the distances between consecutive points.

Let b_i denote the bottom y-coordinate of line segment $i \in \mathcal{C}$, and t_i denote its top y-coordinate. Compute b_i and t_i for all line segments. Sort the list of all values b_i and t_i and define a horizontal slice as the set of all horizontal lines between consecutive values in this sorted list.

Sort the list of line segments in each horizontal slice. By processing the horizontal slices from top to bottom this step can be done in $O(n^2)$ time.

If $|\mathcal{T}| = 1$ then all we need to do is determine whether any of the slices intersect with a single segment in $\mathcal{C}$.

On the other hand if $|\mathcal{T}| \geq 2$ then we proceed as follows. In each horizontal slice containing at least two segments, find two neighbouring line segments that are non-parallel, line segments l_i and l_{i+1} say. Let δ_0 be the distance between the corresponding points p_i and p_{i+1} in $\mathcal{T}$ and let δ_1 be the distance between the corresponding points p_{k-1-i} and p_{k-2-i}. Determine the (at most two) positions of the sectioning line with the correct distance δ_0 or δ_1 between l_i and l_{i+1}. Verify in linear time whether or not the candidate sectioning lines are equivalent to $\mathcal{T}$. If all line segments in the horizontal slice are parallel, then either all horizontal lines in this slice are correct sectioning lines, or none are, which can be verified in linear time.

For each horizontal line on the boundary of a horizontal slice, verify in linear time whether or not it is equivalent to $\mathcal{T}$.

Since there are a linear number of horizontal slices, this algorithm requires $O(n^2)$ time. If some of the line segments are horizontal and $\mathcal{T}$ contains one of them, the above algorithm can be modified and will still require $O(n^2)$ time. $\square$

Acknowledgement

This research was initiated at Bellairs Research Institute of McGill University during the International Workshop on Computational Geometric Tomography.

References

[1] P. Bose, F. Hurtado, H. Meijer, S. Ramaswami, D. Rappaport, V. Sacristán, T. Shermer and G. Toussaint, Finding Specified Sections of Arrangements: 3D Results, manuscript in preparation.

[2] R. J. Gardner, *Geometric Tomography*, Encyclopedia of Mathematics and its Applications, vol. 58, Cambridge University Press, New York, 1995.

[3] G. T. Herman, A. K. Louis and F. Natterer (Eds.), *Mathematical Methods in Tomography*, Lecture Notes in Mathematics, vol. 1497, Springer-Verlag, Heidelberg, 1990.