Controlled Perturbation of Arrangements of Line Segments in $\mathbb{R}^2$ *

Eli Packer,
School of Computer Science, Stony Brook University

Abstract

We present a controlled perturbation scheme for arrangements of line segments in $\mathbb{R}^2$ (CPAL for short). Given an arrangement of line segments in $\mathbb{R}^2$, CPAL perturbs some of the segments to form robust representation, while preserving all topological incidences among the vertices. We developed and implemented several heuristics that come to decrease the perturbation magnitude. The idea is to process the input vertices in a specific order decided by the heuristics.

1 Introduction

Finite-Precision Approximation is an approach for converting input to make it more robust for finite-precision manipulation. The goal is to make the representation more robust to errors induced by the lack of infinite precision.

Controlled Perturbation is a Finite-Precision approximation technique. Its main goal is to eliminate all *potential degeneracies* (those are degeneracies that cannot be realized with floating-point, thus referred as potential), by perturbing the input.

A typical Controlled Perturbation algorithm proceeds as follows. Each element in the input is processed in its turn. It is perturbed if it induces potential degeneracies with elements that have already been processed. The perturbation is done randomly inside a ball with radius δ, called the *perturbation radius*. δ is large enough to guarantee a reasonable probability for finding a valid placement (a placement that induces no potential degeneracies). Then one or more perturbations are performed until a degeneracy-free placement is obtained. Since the probability of finding good placement is reasonable, after few trials on the average, a degeneracy-free placement is obtained.

In order to test potential degeneracies, *resolution parameters* (one or more) have to be set. They define the minimum separations required between geometric features to maintain robust computation. Since different geometric queries may be applied to the underlined arrangement, different resolution parameters are necessary. In the full paper, we compute the various perturbation radii and resolution parameters.

We propose a perturbation scheme for arrangements of line segments in $\mathbb{R}^2$. The input consist of vertices of the arrangement and an incidence list among the vertices that defines the edges of the arrangement. Our scheme will preserve these incidences and separate other features to eliminate potential degeneracies. Preserving incidences will allow us to preserve the topology of certain geometric shapes such as polygons. In this framework, degeneracies may occur at vertices of the arrangement, namely at endpoints or at intersections of edges. The goal is to perturb some of the endpoints enough, such that any edge is sufficiently separated from any non-incident vertex. Note that degeneracies may occur at edge endpoints if the corresponding vertex is incident to more than one segment. However, such degeneracies are definite and can be realized in a robust way. As a result, any potential degeneracy in the output will be categorized as a robust definite degeneracy.

We deal with three types of degeneracies: *endpoint - edge*, *intersection point - edge* and *two endpoints*. Degeneracies occur when the features of each pair are not sufficiently far.

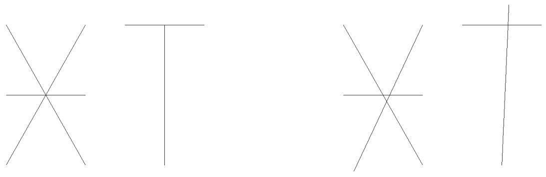
Previous research that used Controlled Perturbation are [1, 3, 4, 5].

2 Algorithm

We propose a framework that works on arrangements of line segments in $\mathbb{R}^2$ (CPAL for short). CPAL follows the framework described in Section 1. Let $S(V, C)$ be the input where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of endpoints in the input and C is the incidences among the endpoints of V, which defines the edges of the arrangement, $A(S)$. The idea is that after perturbing the input, the incidences described in C will be preserved. All edges that are not incident to $v \in V$ according to C, will be sufficiently separated from it. Also, each edge will be sufficiently separated from any intersection point of two other edges. We will achieve these separations by incrementally processing endpoints.

As will be clarified in the full paper, we need to divide our work into two phases (each will use the ideas described above). In the first, we need to sufficiently separate each pair of endpoints, $v_i, v_j \in V$. We denote the perturbation radius of this phase by δ_1 and the resolution parameter (minimum separation between endpoints after perturbation) by ρ_1. In the second phase, we again

*Work on this paper has been partially supported by the National Science Foundation (CCR-0098172, CCF-0431030).



(a) (b)

Figure 1: The input (a) is transformed into (b) by perturbing endpoints to remove potential degeneracies

perturb endpoints, but this time we are concerned in potential degeneracies induced by edges and vertices of $A(S)$. Each endpoint $v \in V$ will be processed by checking whether potential degeneracies occur. Potential degeneracies may be induced both by edges that are too close to v or vertices in the temporary arrangement that are too close to edges incident to v. Once a valid placement for v is found, its location will be fixed and it will not be perturbed again. We denote the perturbation radius of this phase by δ_2. There are four resolution parameters associated with this phase, $\rho_2 - \rho_5$. In the full paper, we compute both the perturbation radii (δ_1 and δ_2) and the resolution parameters ($\rho_1 - \rho_5$), by analyzing the various potential degeneracies in detail.

We use two optimizations to make the work more efficient. The first one restricts the possible candidates that have to be tested for potential degeneracies. The idea is to analyze the input and associate each feature to be tested with only edges and vertices that have a chance to induce potential degeneracies with it. The second optimization tries to decrease the perturbation magnitude in order to have better approximation. The idea is to start with smaller perturbation radii than δ_1 and δ_2 (as δ_1 and δ_2 reflect extreme congested input set), trying to find degeneracy-free placements. Then continue by gradually increasing the perturbation radii until a degenerate-free placement is found. We implemented both optimizations, used them in our experiments and discuss them in the full paper in detail.

Figure 1 illustrates the results of CPAL on a small input set.

3 Ordering the Endpoints

The main drawback of the Controlled Perturbation scheme is clearly the possibility of large endpoint deviations. Thus, it would be desirable to constrain the deviation as much as possible. We note that different endpoint processing orders may result in different forbidden areas (areas for which the endpoint induces degeneracies, if placed inside). The reason is that potential degeneracies involve only features that have already been processed by the time a certain endpoint is processed. Thus, different orderings induce different forbidden areas.

While previous Controlled Perturbation schemes used an arbitrary processing order, we try to develop and use efficient heuristics that improve the quality of the output (or in other words, reduce the perturbations of the endpoints). The two objectives we use are to minimize the maximum perturbation (MIN-MAX) and to minimize the perturbation sum (MIN-SUM).

We build a directed graph $G(V, E)$ on the endpoints and their incidences. Any $e(v_1, v_2) \in E$, is associated with a weight that reflects how much better it is to have v_1 before v_2 in the vertex order processing, $\Pi(G)$. Thus, it is desired avoid as many edges that do not comply with $\Pi(G)$ (we call them *backedges*). MIN-MAX will try to minimize the maximum weight on a backedge while MIN-SUM will try to minimize the sum of the weights of the backedges.

It turns out that the MIN-MAX problem can be efficiently solved and that the MIN-SUM is a hard problem as it can be formulated as a *Feedback Arc Set* problem which is NP-complete [2]. In the full paper, we give an algorithm for the MIN-MAX problem and also show that if negative weights are allowed, the problem is NP-complete. We then describe several available approximation algorithms related to MIN-SUM, algorithms that we implemented and used for experiments. We note that all the above algorithms were useful in many cases. Many experiments we performed showed that the ordering algorithms that we used decreased the average perturbation magnitudes by 50% and more.

References

[1] S. Funke, C. Klein, K. Mehlhorn, and S. Schmitt. Controlled perturbation for delaunay triangulations. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1047–1056, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[2] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman And Company, New York, 1979.

[3] D. Halperin and E. Leiserowitz. Controlled perturbation for arrangements of circles. *International Journal of Computational Geometry and Applications*, 14(4 and 5):277–310, 2004. Special issue, papers from SoCG 2003.

[4] D. Halperin and C. R. Shelton. A perturbation scheme for spherical arrangements with application to molecular modeling. *Comput. Geom. Theory Appl.*, 10:273–287, 1998.

[5] S. Raab. Controlled perturbation of arrangements of polyhedral surfaces with application to swept volumes. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 163–172, 1999.