Controlled Perturbation for Delaunay Triangulations

slides prepared by Christian Klein for SODA 2005,

extended for lectures on algorithm engineering by Kurt Mehlhorn

Stefan Funke, Christian Klein, Kurt Mehlhorn, Susanne Schmitt

Max Planck Institut für Informatik, Saarbrücken

Overview

- Idealistic Algorithms
- Controlled Perturbation
- Delaunay Triangulations
- General Randomized Construction
- Experimental Results

Idealistic Algorithms

Simplifying assumptions when designing computational geometry algorithms:

- The Real RAM Assumption (we can compute with reals)

  Only have fixed precision floating point arithmetic!

  Implementing Real RAM is costly!

- The Non-Degeneracy Assumption, e.g. no 3 collinear points, no 4 cocircular points

  Occur in practice!

  Need High Precision to detect!

  Handling all Degeneracies complicates code!

Predicates

Geometric algorithms base decisions on predicates.

Example: Given $(d+1)$ points $p_0, \ldots, p_d \in \mathbb{R}^d$, the orientation test gives the position of p_d relative to the hyperplane spanned by $p_0, \ldots, p_{d-1}$.

$$orient(p_0, \ldots, p_d) := \operatorname{sign} \begin{vmatrix} p_{01} & \cdots & p_{0d} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ p_{d1} & \cdots & p_{dd} & 1 \end{vmatrix}.$$

Evaluates to zero iff all points lie in a common hyperplane.

$\rightsquigarrow$ Degeneracy

The Controlled Perturbation Framework

Overview

- Solve problem not for given input but for "nearby" perturbed input.
- Choose perturbation carefully: such that the perturbed input
  - is in general position.
  - can be solved with fixed precision arithmetic.

⇝ Can run idealistic algorithm on perturbed input.

The sales pitch for controlled perturbation: it numerically perturbs the inputs such that they can be processed correctly and fast by simple algorithms.

Previous Work

Controlled Perturbation was proposed by Halperin et. al. and applied to

- spherical arrangements [Halperin, Shelton 1998]
- polyhedral arrangements [Halperin, Raab 1999]
- arrangements of circles [Halperin, Leiserowitz 2003].

- We describe a general framework (guard predicates) for controlled perturbation,
- apply it to Delaunay triangulations and convex hulls in any dimension,
- and analyze the use of controlled perturbation in randomized algorithms.

Guard Predicates

To guard an expression E, e.g., the expression for the orientation predicate, against round-off errors, we need a guard predicate $\mathcal{G}_E$:

If $\mathcal{G}_E$ evaluates to true when evaluated with floating point arithmetic, the evaluation of E with floating point arithmetic yields the correct sign.

use error bounds [e.g. Mehlhorn, Näher] to derive $\mathcal{G}_E$, namely

- replace branch on sign of E by
- if $|E| > B$ (as computed in last lecture) branch on sign of E else abort

Modify idealistic algorithm A as follows:

- Guard each predicate E.
- If any guard fails abort algorithm, else report result.

$\rightsquigarrow$ Guarded Algorithm A_g.

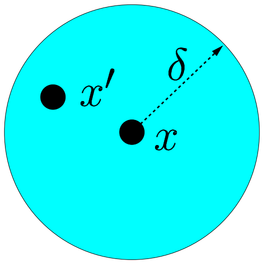
Fact: A_g follows execution path of A and has same running time.

Controlled Perturbation: The Basic Scheme

Controlled Perturbation Scheme:

Input: $S = \{x_1, \ldots, x_n\}$, Perturbation value $\delta > 0$

* Compute random δ-perturbation S' of S, i.e., replace each point x by a random point x' in the δ-disc centered at x.
* Run A_g on S'.
* If A_g fails, restart.
* Report result.



Question: What is a good value of δ? Answer will depend on the precision p of floating point computation (this was called ϵ in the last lecture) and input, e.g. the maximum coordinates M.

Controlled Perturbation: Usage Szenarios

MAX-PLANCK-GESELLSCHAFT

- Usage 1: keep the precision p fixed
  - set δ to some initial value δ_0
  - run controlled perturbation up to c times
  - if all executions abort, double δ and repeat
- Usge 2: keep δ fixed
  - set p to some initial value p_0, e.g., 52 (double precision floats)
  - run controlled perturbation up to c times
  - if all executions abort, double p and repeat
- what is the appropriate choice of c?
- can we expect reasonable values of δ and p?
- second question can be answered experimentally and analytically

On the Choice of c?

- in the second szenario (fixed δ, variable p), the running time will depend on p, because the cost of arithmetic will depend on the mantissa length in use.
- assume running time is $p^k \cdot T$, where T depends on the input but not on p.
- if only addition and multiplications and school method is used, $k = 2$.
- let $f(p)$ be the property of abortion when precision p is used. We assume that $f(p)$ decreases and goes to zero as p increases
- let p^* be such that $f(p^*) \le 1/2$. $\quad p^* = 2^{i^*} p_0$
- expected running time is bounded by

$$\sum_{i \ge 0} \Big(\prod_{j<i} f(2^j p_0)^c\Big) \cdot c \cdot (2^i p_0)^k \cdot T = \sum_{i \le i^*} c(2^i p_0)^k T + \sum_{i > i^*} (1/2)^{c(i-i^*)} \cdot c(2^i p_0)^k T$$

$$\le c(2^{i^*} p_0)^k T + c(2^{i^*} p_0)^k T \sum_{i \ge i^*} \frac{2^{k(i-i^*)}}{2^{c(i-i^*)}} = O(c(p^*)^k T) \quad \text{provided that } c > k$$

Randomized Algorithms

Let A be a randomized algorithm .

Theorem: If δ is such that A_g has constant failure probability $1/c$ on a δ-perturbed input, then the expected asymptotic running time of controlled perturbation is the running time of A.

see next slide for a proof

Application to RICs: RICs process the n input objects one by one. In order to guarantee an overall failure probability $1/c$, we need failure probability $1/cn$ for any of the n insertions.

Randomized Algorithms

Let A be a randomized algorithm,

$T(x, \pi)$ its running time on input x with random bits $\pi \in \{0,1\}^m$,

$T(x) = 2^{-m} \sum_\pi T(x, \pi)$ its expected running time on input x

$U_\delta(x)$ the set of all δ-perturbations of x

$T_\delta(x) = \mathrm{E}_{x' \in U_\delta(x)}[T(x')] = \frac{1}{|U_\delta(x)|} \sum_{x' \in U_\delta(x)} T(x')$ the δ-smoothed running time at x.

For A_g we have $T_g(x, \pi) = \mathcal{O}(T(x, \pi))$ for all x and π.

Let $\chi(x, \pi) = 1$ if $A_g(x, \pi)$ aborts, zero otherwise. Then

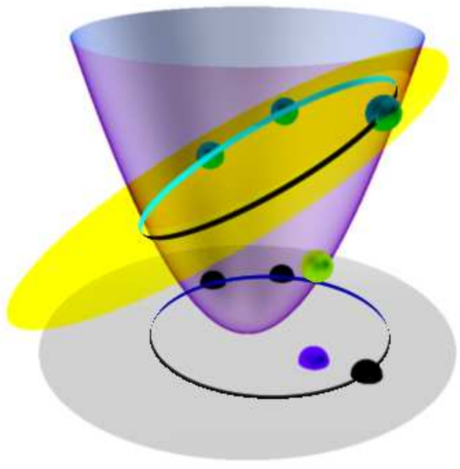
$$p_\delta(x) = \sum_\pi \sum_{x' \in U_\delta(x)} \frac{\chi(x', \pi)}{2^m \cdot |U_\delta(x)|}$$

is the probability that A_g fails on a random $x' \in U_\delta(x)$

running time T_g of A_g on x: $T_g(x) = T_\delta(x) + p_\delta(x) T_g(x) = \frac{T_\delta(x)}{1 - p_\delta(x)}$.

Predicates and Guards

Delaunay RIC uses predicates $orient(p, q, r)$ and $incircle(p, q, r, t)$,



where $incircle(p, q, r, t)$
$= orient(l(p), l(q), l(r), l(t))$
with $l((x, y)) = (x, y, x^2 + y^2)$

Can derive error bounds

$$B_{Orient} = 24 \cdot M^2 2^{-p}, \quad B_{Incircle} = 432 \cdot M^4 2^{-p}.$$

Guards ensure predicates have absolute values greater than error bounds.

Delaunay Triangulations I

A triangulation is called Delaunay triangulation if the interior of the circumcircle of any triangle contains no point.

RIC for Delaunay triangulation in $\mathcal{O}(n \log n)$:

- Start with infinite Triangle.
- Pick a new point p at random.
- Use history to locate triangle containing p.
- Split Triangle and “legalize” triangulation.

Expected number of edges generated during the algorithm is $6n$ and at most twice as many triangles are generated.

for details of the algorithm see LEDAbook or deBerg/vanKreveld/Overmars/Schwarzkopf or ...

Legalize Triangulation

MAX-PLANCK-GESELLSCHAFT

- initialize a list L to the three new triangles with vertex p
- invariant: all triangles in L will have p as a vertex
- while L is non-empty do
  - let (p, a, b) be one of the triangles in p.
  - let (a, b, c) the other triangle sharing the edge (a, b)
  - if (p, a, b, c) form a convex quadrilateral and p lies inside the circumcircle of (a, b, c), replace the triangles (p, a, b) and (a, b, c) by the triangles (p, a, c) and (p, b, c). Put the new triangles on L.

- connection to 3d convex hull of lifted points

Experimental Results

Input, pts	avg. δ (lazy)	avg. δ (standard)
Flower, 400	0.05877	0.001
Flower, 2000	0.20473	0.0023
Flower, 10000	0.79845	51.76
Grid, 441	0.00308	0.001
Grid, 2601	0.00675	0.0043
Grid, 10201	0.01299	0.105
Grid, 160801	0.05181	>grid

lazy: choose δ so that alg works and perturb points only if necessary

standard: use δ as given by analysis and perturb every point

Controlled Perturbation for Delaunay Triangulations

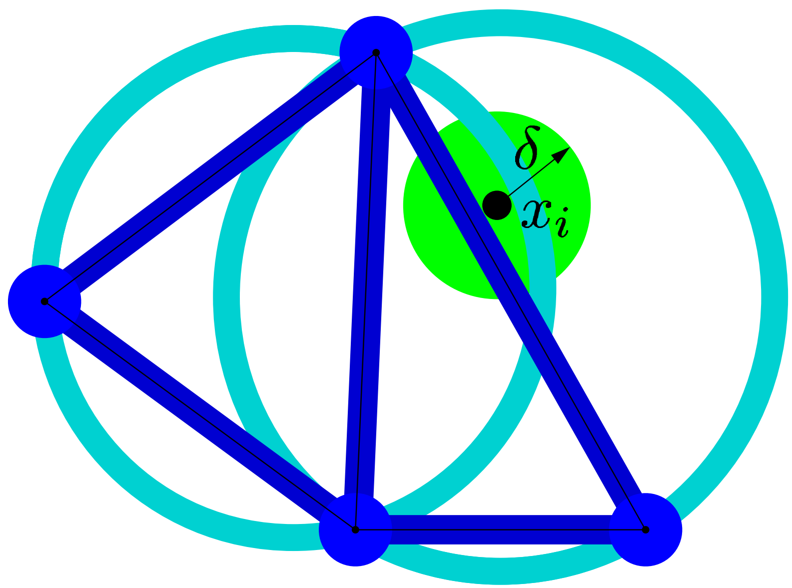
The Analysis

Delaunay Triangulations

Want δ such that guards hold with reasonable probability when inserting point x_i^{pert}.

Areas where guards fail: Forbidden Areas.

$\rightsquigarrow$ only a small fraction of δ-disc should be forbidden; more precisely, a fraction $1/(cn)$. This guarantees failure probability $1/(cn)$ for single insertion and $1/c$ overall. Concrete, $c = 2$.



Delaunay Triangulations

Geometrically, guards hold when evaluating

$\rightsquigarrow$ $orient(p, q, x_i)$, if points form triangle of area at least $B_{Orient}/2$.

since $orient(p, q, x_i) =$ signed area of triangle $\Delta(p, q, x_i)$

$\rightsquigarrow$ $incircle(p, q, r, x_i)$, if x_i lies outside $B_{Incircle}/\mathrm{area}(\Delta)^{3/2}$-annulus around circumcircle of $\Delta(p, q, r)$.

since $incircle(p, q, r, x_i) \geq \mathrm{area}(\Delta)^{3/2} \cdot \mathrm{dist}(x_i, Circumcircle)$

We further require (as a kind of induction hypothesis)

- $\mathrm{dist}(p, q) \geq \xi$ for all points
- $\mathrm{area}(\Delta(p, q, r)) \geq \xi_\Delta > B_{Orient}/2$ for all created triangles.

Delaunay Triangulations

Geometrically, guards hold when evaluating

- $\rightsquigarrow$ $orient(p,q,x_i)$, if points form triangle of area at least $B_{Orient}/2$.

  since $orient(p,q,x_i) =$ signed area of triangle $\Delta(p,q,x_i)$

- $\rightsquigarrow$ $incircle(p,q,r,x_i)$, if x_i lies outside $B_{Incircle}/\mathrm{area}(\Delta)^{3/2}$-annulus around circumcircle of $\Delta(p,q,r)$.

  since $incircle(p,q,r,x_i) \geq \mathrm{area}(\Delta)^{3/2} \cdot \mathrm{dist}(x_i, Circumcircle)$

We further require (as a kind of induction hypothesis)

- $\mathrm{dist}(p,q) \geq \xi$ for all points
- $\mathrm{area}(\Delta(p,q,r)) \geq \xi_\Delta > B_{Orient}/2$ for all created triangles.

Delaunay Triangulations

- average number of edges constructed is at most $6n$
- more than $24n$ edges with probability at most $1/4$
- we restrict to executions with at most $24n$ constructed edges

- each point generates a forbidden region of size $\pi\xi^2$
- total forbidden region due to points: $n\pi\xi^2$

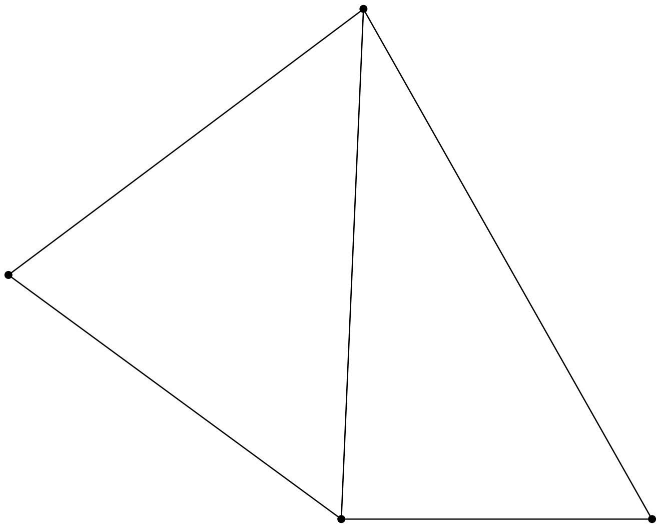
- each edge e generates a forbidden region
  - length of edge is at least ξ
  - no point in strip of half-width $2\xi_\Delta/\xi$ around $\ell(e)$ guarantees that all triangles (e, x) have area at least ξ_Δ
  - strip intersects disk of radius δ in a region of area at most $2\delta 4\xi_\Delta/\xi$
  - total forbidden region due to edges: $24n8\delta\xi_\Delta/\xi$

Delaunay Triangulations

Lemma: If

$$\pi\delta^2/(4n) \geq (\qquad\qquad\qquad\qquad),$$

the guarded Delaunay algorithm will succeed with probability at least $1/2$.

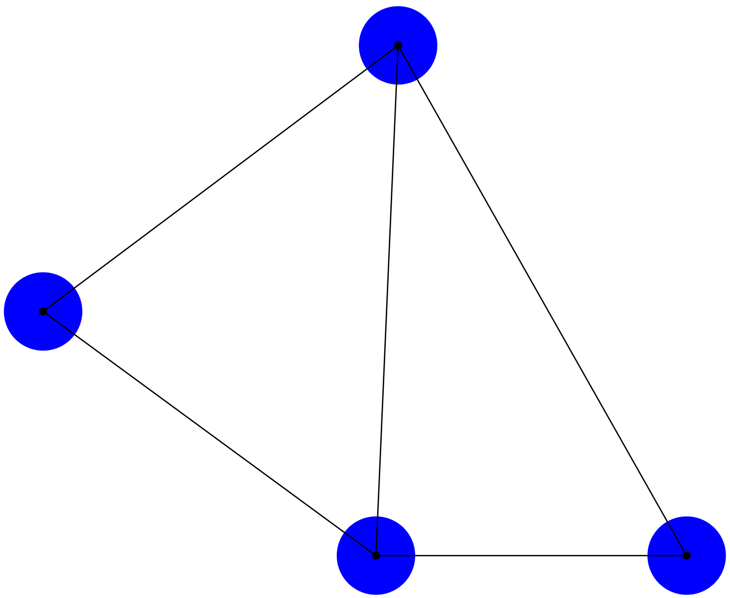


Delaunay Triangulations

Lemma: If

$$\pi\delta^2/(4n) \geq (\mathbf{n}\pi\xi^2 \qquad),$$

the guarded Delaunay algorithm will succeed with probability at least $1/2$.

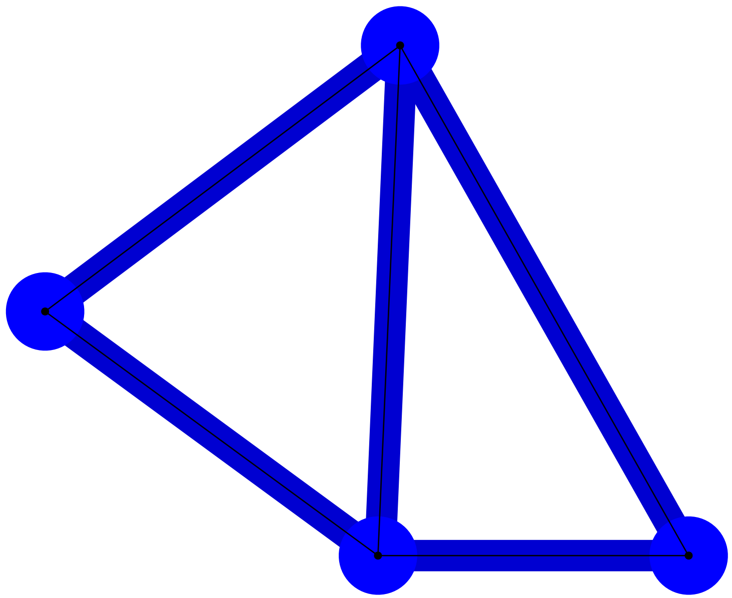


Delaunay Triangulations

Lemma: If

$$\pi\delta^2/(4n) \geq (\mathbf{n}\pi\xi^2 + \mathbf{24n} \cdot 8\delta\xi_\Delta/\xi \qquad),$$

the guarded Delaunay algorithm will succeed with probability at least $1/2$.

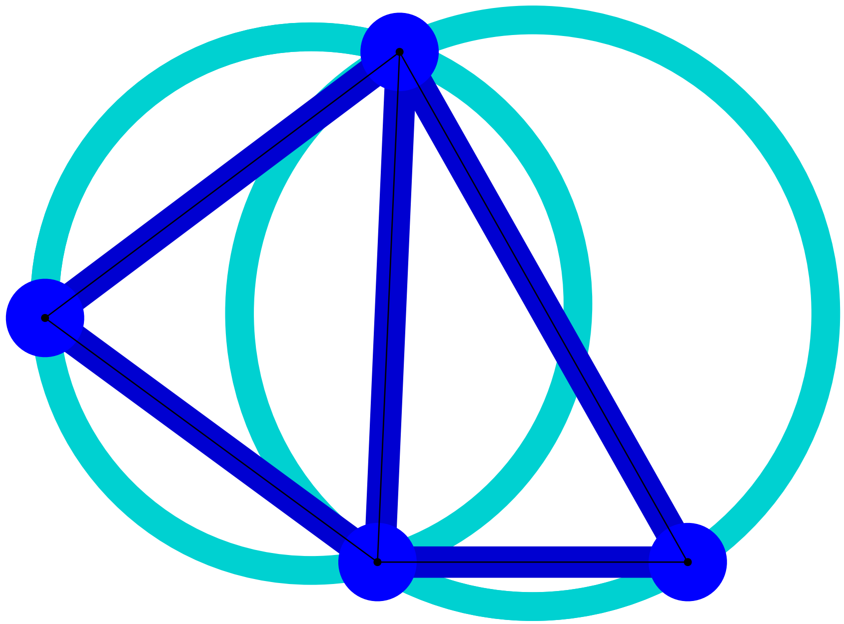


Delaunay Triangulations

Lemma: If

$$\pi\delta^2/(4n) \geq (\mathbf{n}\pi\xi^2 + \mathbf{24n} \cdot 8\delta\xi_\Delta/\xi + \mathbf{2n} \cdot 4\pi\delta B_{Incircle}/\xi_\Delta^{3/2}),$$

the guarded Delaunay algorithm will succeed with probability at least $1/2$.



Delaunay Triangulations

Lemma: If

$$\pi\delta^2/(4n) \geq (\mathbf{n}\pi\xi^2 + \mathbf{24n} \cdot 8\delta\xi_\Delta/\xi + \mathbf{2n} \cdot 4\pi\delta B_{Incircle}/\xi_\Delta^{3/2}),$$

the guarded Delaunay algorithm will succeed with probability at least $1/2$.

Theorem: If the guarded algorithm is executed with precision p, where $p \geq C(\log M - \log\delta + \log n + 1)$ for a suitable constant C, it succeeds with probability at least 1/2.

Open Problems

- apply approach to problems which require non-rational arithmetic
- Voronoi diagrams of line segments
- Arrangements of Conics
- Arrangements of High Degree Curves