Delaunay Triangulation of Imprecise Points

Preprocess and actually get a fast query time

Ján Bella Maxime Portaz

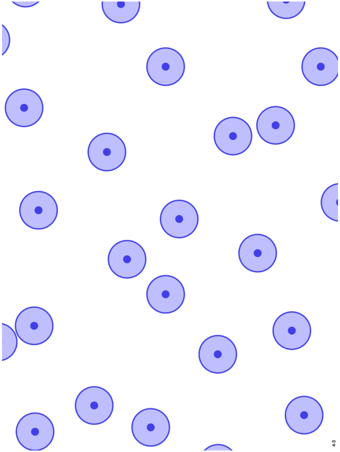
Grenoble INP

January 23, 2015

Problem

Given: a set of regions (imprecise points)

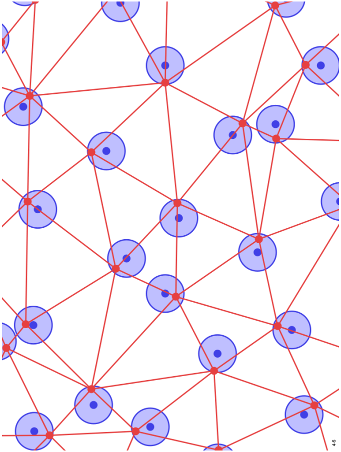
Is there an advantage we can take and find Delaunay triangulation effectively?



Problem

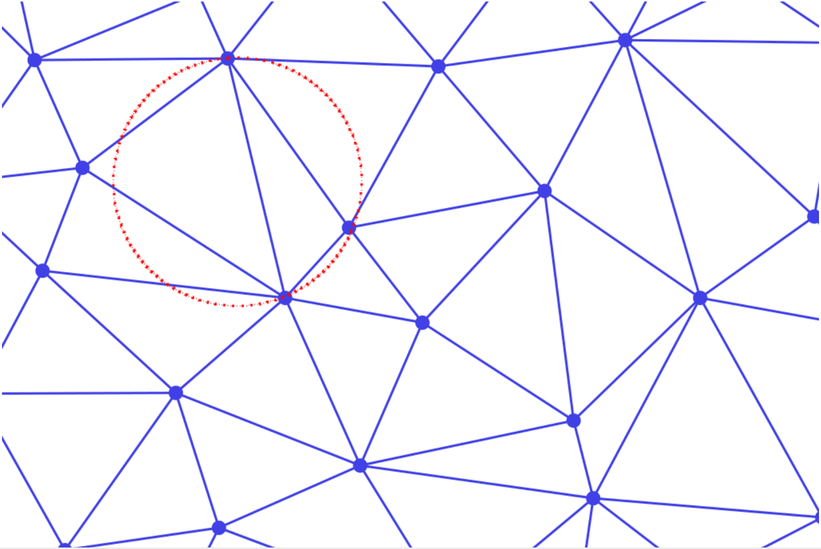
Given: a set of regions (imprecise points)

Is there an advantage we can take and find Delaunay triangulation effectively?



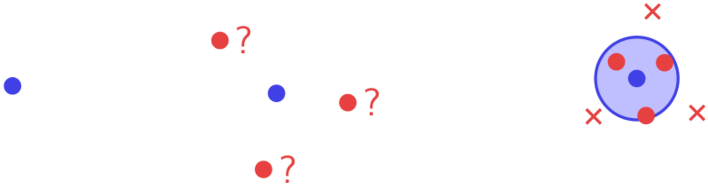
Outline

Delaunay Triangulation



Imprecise point

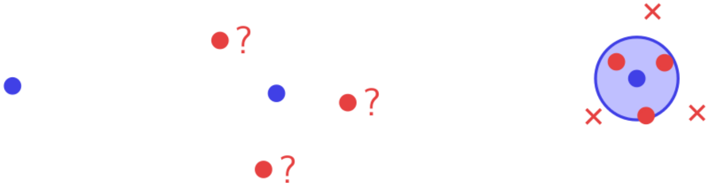
Imprecise point: extending a point to some region



Supposing we have precise locations within the regions (point instances)

Imprecise point

Imprecise point: extending a point to some region



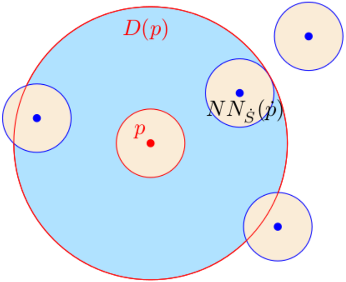
Supposing we have precise locations within the regions (point instances)

Notation

Imprecise point notation

- $DT_P \rightarrow$ Delaunay triangulation of set of points P
- $NN_P(v) \rightarrow$ nearest neighbour of point $v \in P$ in $P \setminus \{v\}$
- $\dot{p} \rightarrow$ the center of imprecise point p
- $\hat{p} \rightarrow$ an instance of imprecise point p.
- $S, \dot{S}, \hat{S}$ analogously being the sets of points

Disk



$D(p) \rightarrow$ the disk with center p and radius $||\dot{p}NN\dot{S}(\dot{p})|| + 1$, in case of disjoint unit disks S

Imprecise points

Preprocess unit disks in $O(n\ log\ n)$ and compute Delaunay triangulation in $O(n)$

- Maarten Löffler and Jack Snoeyink. 2008. Delaunay triangulations of imprecise points in linear time after preprocessing
  → mainly theoretical
- Buchin et al. 2011. Preprocessing Imprecise Points for Delaunay Triangulation: Simplified and Extended
  → heavy in practice (Delaunay triangulation of 8n points)

Delaunay construction

Walking strategy

Straight walk in the triangulation[1]

1. Visit every triangles crossed by the line between the starting point a the new point
2. Remove non-Delaunay triangles
3. Filling hole with triangles incident to the new point

[1] Olivier Devillers. Walking in a Triangulation. Proceedings of the Seventeenth Annual Symposium on Computational Geometry, 201,9, pp.106-114

Preprocessing

Steps

1. $S = \{p_1, p_2, ..., p_n\}$ enumerate disks in a random order
2. Compute $DT_{\dot{S}}$ incrementally, inserting points in the order of their indices
3. For each $\dot{p}_k$ inserted, we compute the hint $h(k)$ of p_k
   $\rightarrow$ hint : $NN_{\dot{S}_k}(\dot{p}_k) = \dot{p}_{h(k)}$

Expected time

Randomized incremental construction $\rightarrow O(n\ log\ n)$[2]
Including the computation of $h(k)$ for each k

[2] Olivier Devillers. The Delaunay Hierarchy. International Journal of Foundations of Computer Science, World Scientific Publishing, 2002, 13, pp.163-180

Instance processing

Steps

- Given an instance $\hat{S}$ we compute $DT_{\hat{S}}$ inserting points in the order of their indices
- Location of $\hat{p}_k$ in $DT_{\hat{S}_{k-1}}$ is done by straight walk starting at $\hat{P}_{h(k)}$

Expected time

$O(n) \rightarrow$ better in practice

Complexity

Unit disjoint regions

Expected cost of building $D_{\hat{S}}$ is linear.

Proof

By backward analysis, the cost of adding last point $\hat{p}$ is constant: 3 steps:

- visit triangles incident to $\hat{x} \in DT_{\hat{S} \setminus \{\hat{p}\}}$
- visit triangles crossed by line segment $\hat{x}\hat{p}(\dot{x} = NN_{\dot{S}}(\dot{p}))$
- update triangulation to $DT_{\hat{S}}$

Partial costs

visit triangles incident to $\hat{x}$

$$d^\circ DT_{\hat{S}\setminus\{\hat{p}\}}(\hat{x}) \leq d^\circ{}_{DT_{\hat{S}}}(\hat{x}) + d^\circ{}_{DT_{\hat{S}}}(\hat{p})$$

visit triangles crossed by line segment $\hat{x}\hat{p}$

- edges that disappear in $DT_{\hat{S}}$
  $\rightarrow d^\circ{}_{DT_{\hat{S}}}(\hat{p}) - 3$
- edges in $d^\circ{}_{DT_{\hat{S}}}$ crossed by $\hat{x}\hat{p}$
  $\rightarrow \sum_{\hat{q}\in D(p)} d^\circ{}_{DT_{\hat{S}}}(\hat{p})$ [will be proved later]

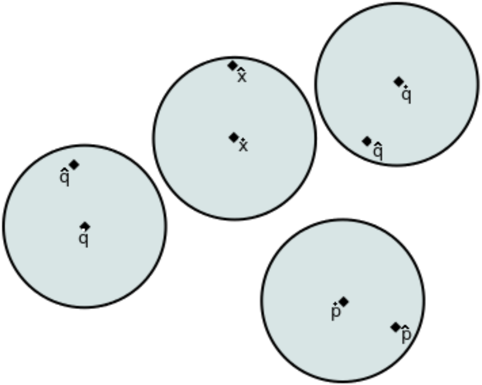
update triangulation to $DT_{\hat{S}}$

$$d^\circ{}_{DT_{\hat{S}}}(\hat{p})$$

Lemma

Statement

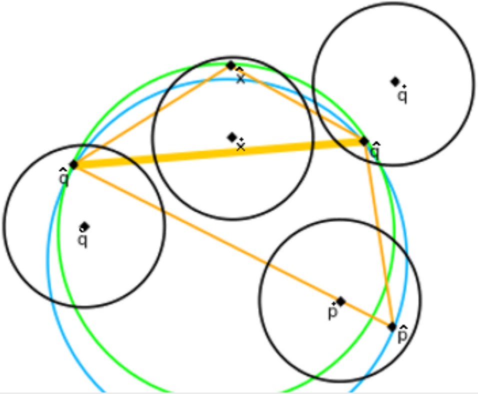
If edge $\hat{q}\hat{q}'$ of $DT_{\hat{S}}$ intersects line segment $\hat{x}\hat{p}$ with $\dot{x} = NN_{\dot{S}}(\dot{p})$, then either $\hat{q}$ or $\hat{q}'$ belongs to $D(p)$.



Lemma

Statement

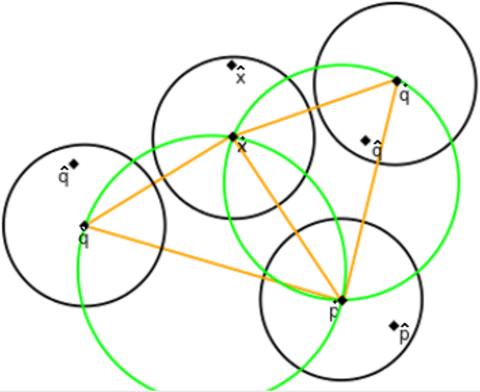
If edge $\hat{q}\hat{q}\prime$ of $DT_{\hat{S}}$ intersects line segment $\hat{x}\hat{p}$ with $\dot{x} = NN_{\dot{S}}(\dot{p})$, then either $\hat{q}$ or $\hat{q}\prime$ belongs to $D(p)$.



Lemma

Statement

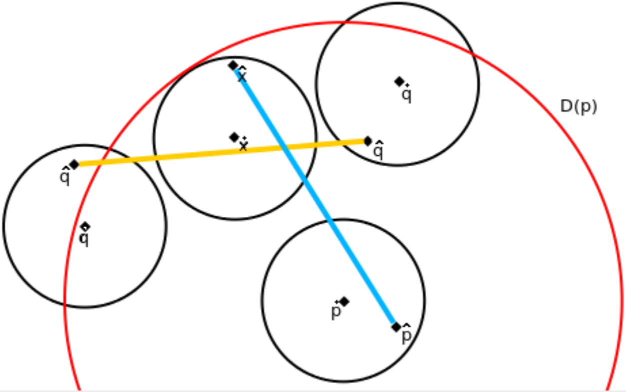
If edge $\hat{q}\hat{q}\prime$ of $DT_{\hat{S}}$ intersects line segment $\hat{x}\hat{p}$ with $\dot{x} = NN_{\dot{S}}(\dot{p})$, then either $\hat{q}$ or $\hat{q}\prime$ belongs to $D(p)$.



Lemma

Statement

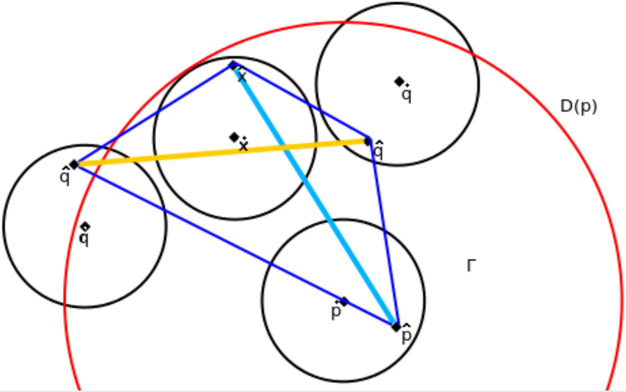
If edge $\hat{q}\hat{q}\prime$ of $DT_{\hat{S}}$ intersects line segment $\hat{x}\hat{p}$ with $\dot{x} = NN_{\dot{S}}(\dot{p})$, then either $\hat{q}$ or $\hat{q}\prime$ belongs to $D(p)$.



Lemma

Statement

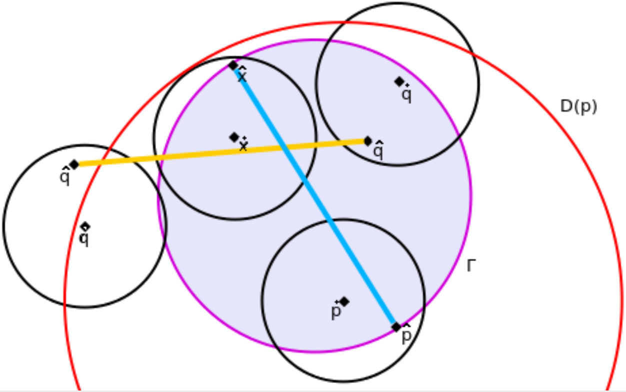
If edge $\hat{q}\hat{q}\prime$ of $DT_{\hat{S}}$ intersects line segment $\hat{x}\hat{p}$ with $\dot{x} = NN_{\dot{S}}(\dot{p})$, then either $\hat{q}$ or $\hat{q}\prime$ belongs to $D(p)$.



Lemma

Statement

If edge $\hat{q}\hat{q}\prime$ of $DT_{\hat{S}}$ intersects line segment $\hat{x}\hat{p}$ with $\dot{x} = NN_{\dot{S}}(\dot{p})$, then either $\hat{q}$ or $\hat{q}\prime$ belongs to $D(p)$.



Beyond disjoined planar unit disks

Overlapping unit disks

Disks overlapping at most k times: $\mathcal{O}(kn)$

Non-unit disks

Different sizes + non-overlapping: complexity unknown yet
Different sizes, overlapping at most twice: proved quadratic complexity

Higher dimension

$\mathcal{O}(n) + \mathcal{O}(n \log(n))$ preprocessing

Experiment

Compared properties

- Imprecise points vs. classical Delaunay triangulation
- Point sets: Random discs, Brownian motion, Random Balls, 3D noisy data
- running time and # triangles visited

Compared methods

- **Delaunay hierarchy**: allows to insert points dynamically.
- **spatial sort**: points ordered along a curve, then inserted in the spatial sort order.
- **Shewchuk**: based on divide and conquer method.

Running time

2D random imprecise points	running time (μs) per point					
n	10^3	10^4	10^5	10^6	10^7	10^8
spatial sort	1.1	0.85	0.83	0.90	1.0	1.13
Delaunay hierarchy	1.8	1.6	2.8	5.78	9.0	13
Skewchuch	0.96	1.12	1.05	1.61	2.4	
hint random order	0.9	0.88	1.2	2.9	3.8	5.4
hint spatial sort	1.0	0.79	0.59	0.61	0.61	0.62

Running time

2D Brownian motion	running time (μs) per point					
n	10^3	10^4	10^5	10^6	10^7	10^8
spatial sort	0.78	0.78	0.88	0.96	1.12	1.20
hint spatial sort	0.73	0.69	0.79	0.81	0.83	0.82

Running time

3D random imprecise points	running time (μs) per point				
n	10^3	10^4	10^5	10^6	10^7
spatial sort	9.0	7.6	8.0	8.2	8.4
Delaunay hierarchy	11	9.7	18	25	33
hint random order	9.2	7.8	14.2	19	23
hint spatial sort	9.5	7.5	7.8	7.9	8.0

Running time

3D noisy sample of scanned models	running time (μs) per point			
n	10^3	10^4	10^5	full size ($2 \cdot 10^6$ points)
spatial sort	7	8.2	8.6	8.9
hint spatial sort	7	7.5	7.7	7.5

Visited triangles

2D random imprecise points	number of visited triangles per point					
n	10^3	10^4	10^5	10^6	10^7	10^8
spatial sort	3.74	3.64	3.71	3.67	3.55	3.71
Delaunay hierarchy	24	28	29	38	45	47
hint random order	2.83	2.8	2.77	2.75	2.75	2.74
hint spatial sort	2.82	2.80	2.77	2.76	2.75	2.75

Visited triangles

2D Brownian motion	number of visited triangles per time step					
n	10^3	10^4	10^5	10^6	10^7	10^8
spatial sort	3.81	3.68	3.77	3.72	3.62	3.78
hint spatial sort	2.77	2.77	2.77	2.77	2.77	2.77

Visited triangles

3D random imprecise points	number of visited triangles per point				
n	10^3	10^4	10^5	10^6	10^7
hint random order	5.2	5.3	5.3	5.2	5.2
spatial sort	6.3	6.6	6.6	6.6	6.6
Delaunay hierarchy	21	29	34	42	50
hint spatial sort	4.4	4.6	4.5	4.5	4.4

Visited triangles

3D noisy sample of scanned models	number of visited triangles per point			
n	10^3	10^4	10^5	full size ($2 \cdot 10^6$ points)
spatial sort	7.0	8.0	8.6	9.5
hint spatial sort	5.7	6.0	6.1	6.4

Conclusion

Summary

- simple algorithm to describe and to implement
- usable for any type of data
- preprocessing points with spatial sort increases performance rapidly

Do you have any questions?

Thank you for your attention.

Do you have any questions?

Thank you for your attention.

Bibliography

Olivier Devillers.
Delaunay triangulation of imprecise points, preprocess and actually get a fast query time.
Journal of Computational Geometry, 2:30–45, 2011.

Maarten Snoeyink, Jack Löffler.
Delaunay triangulations of imprecise points in linear time after preprocessing.
Available online:
`http://www.staff.science.uu.nl/~loffl001/publications/slides/preprocessing_delaunay.pdf`, 2010.
[Presentation slides].