Lost in Space

Binary search trees beyond dimension one

Jeff Abrahamson

Google, Inc.

The views expressed in these slides are the author's and do not necessarily reflect those of Google.

London Big-O Meetup, 21 May 2014

Outline

$\mathbb{R}^1$

- BST

Outline

$\mathbb{R}^1$

- BST

$\mathbb{R}^d$

- What could go wrong?
- Quadtrees
- kd-trees
- Range trees
- More

Outline

$\mathbb{R}^1$

BST

$\mathbb{R}^d$

What could go wrong?
Quadtrees
kd-trees
Range trees
More

Binary Search Trees

- Is $p \in S$?
- Given x, what is closest point $p \in S$?
- Find $\{x \mid x \in [p_1, p_2]\}$.
- Given δ, find $\{x \mid \mathrm{d}(x, p) < \delta\}$.

Binary Search Trees

- Is $p \in S$?
- Given x, what is closest point $p \in S$?
- Find $\{x \mid x \in [p_1, p_2]\}$.
- Given δ, find $\{x \mid \mathrm{d}(x, p) < \delta\}$.

This is easy in $\mathbb{R}^1$. What about $\mathbb{R}^d$?

Outline

$\mathbb{R}^1$

- BST

$\mathbb{R}^d$

- What could go wrong?
- Quadtrees
- kd-trees
- Range trees
- More

What could go wrong?

The Curse of Dimensionality

What could go wrong?

The Curse of Dimensionality

Some things to think about:

- Volume of unit cube $\pm\epsilon$
- Distance from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$
- Physics: $1/r^{d-1}$

What could go wrong?

The Curse of Dimensionality

Some things to think about:

- Volume of unit cube $\pm\epsilon$
- Distance from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$
- Physics: $1/r^{d-1}$

Richard Ernest Bellman, Dynamic programming, Princeton University Press, 1957.

What could go wrong?

The Curse of Dimensionality

Some things to think about:

- Volume of unit cube $\pm\epsilon$
- Distance from $(0, 0, \ldots, 0)$ to $(1, 1, \ldots, 1)$
- Physics: $1/r^{d-1}$

Richard Ernest Bellman, Dynamic programming, Princeton University Press, 1957.

It's easy to get lost.

Outline

$\mathbb{R}^1$

- BST

$\mathbb{R}^d$

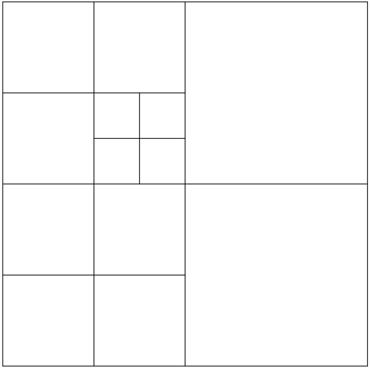
- What could go wrong?
- Quadtrees
- kd-trees
- Range trees
- More

Quadtrees in Words

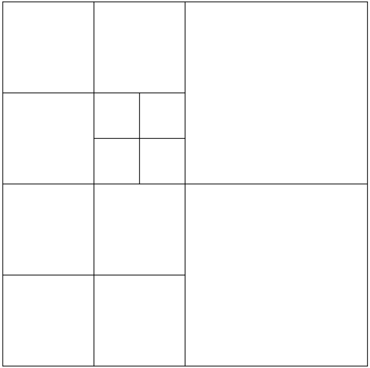
Properties (let's start in $\mathbb{R}^2$)

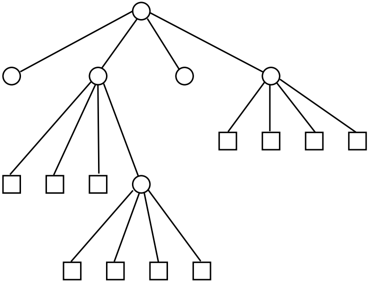
- Rooted tree
- Internal nodes have four children

Quadtrees in Pictures



Quadtrees in Pictures





Quadtrees in Mathematics

- Find a bounding box: $O(n)$
- Then,
  - Divide in four quadrants: $O(1)$
  - Partition the points: $O(n)$
  - Recursively build a quad tree: ??

Quadtrees in Equations

- Depth is at most $\delta = \left\lceil \log\left(\frac{s}{c}\right) + \frac{3}{2} \right\rceil$, where c is the smallest inter-point distance and s is the side length of the initial square.
- A quadtree of depth δ storing n points has $O((\delta + 1)n)$ nodes and requires $O((\delta + 1)n)$ time to construct.
- Finding neighbors in a given direction takes time $O(\delta + 1)$
- Balancing a quadtree takes time $O((\delta + 1)m)$, where m is the number of interior nodes.

Quadtrees in Popular Culture

- One of the first DS's for higher-dimensional data
- Finkel and Bentley, 1974 *(R. A. Finkel and J. L. Bentley, Quad trees: a data structure for retrieval on composite keys. Acta Inform, 4:1–9, 1974)*
- Still perform relatively well
- Easy to generalize to higher dimensions—called octrees

M. de Berg et al., *Computational Geometry: Algorithms and Applications*, second edition, chapter 14. Cf. also chapter end notes.

Outline

$\mathbb{R}^1$

- BST

$\mathbb{R}^d$

- What could go wrong?
- Quadtrees
- kd-trees
- Range trees
- More

Kd-Trees in Words

- Range queries
- Partial match queries
- But not exact match queries (too easy)

Kd-Trees in Words

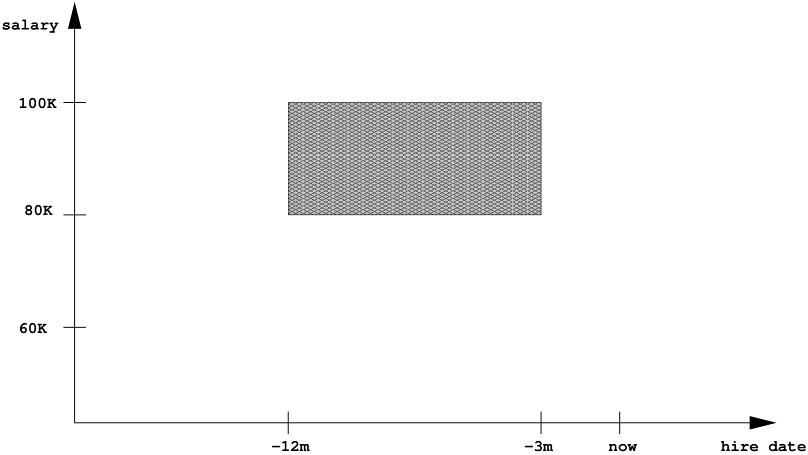
- Range queries
- Partial match queries
- But not exact match queries (too easy)
- Data bases! Records are points in space.

Kd-Trees in SQL

Find employees who are past their trial period but in their first year and who are close to hitting the £100K tax rule start.

```
SELECT salary, hire_date
FROM EMPLOYEES
WHERE 80000 < salary
  AND salary < 100000
AND now − 12m < hire_date
  AND hire_date < now − 3m
```

Kd-Trees in Pictures



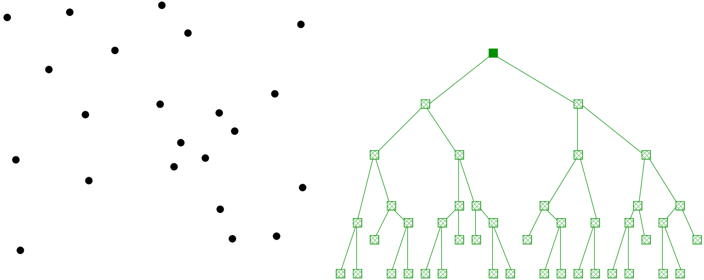
Kd-Trees in $\mathbb{R}^1$

- Binary search tree
- Find beginning of range
- Tree walk to end of range

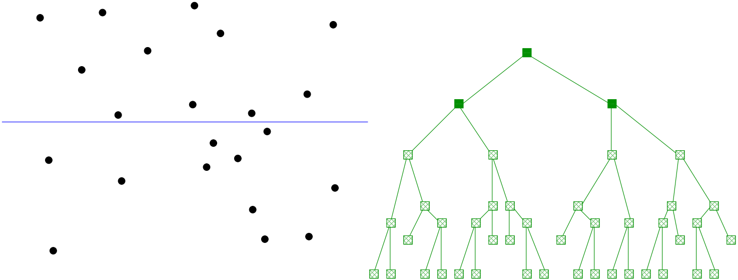
Kd-Trees in $\mathbb{R}^2$

- At root, start with all points.
- Partition on x value to make level 1 nodes
- Partition those sets on y value to make level 2 nodes
- Repeat until done
- Result of a query is a forest of subtrees

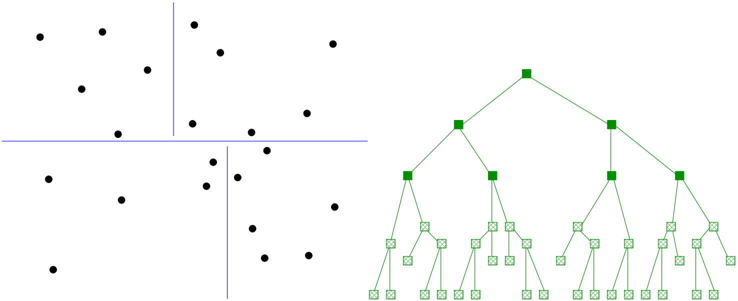
Kd-Trees in Pictures



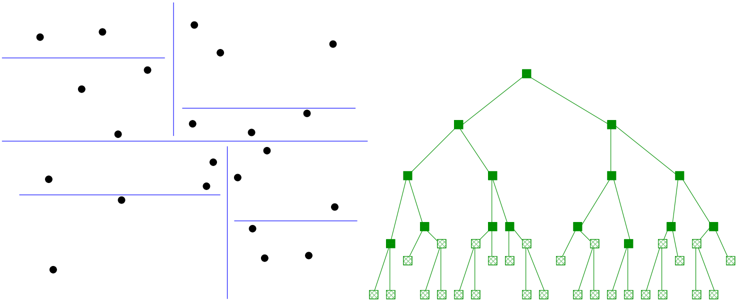
Kd-Trees in Pictures



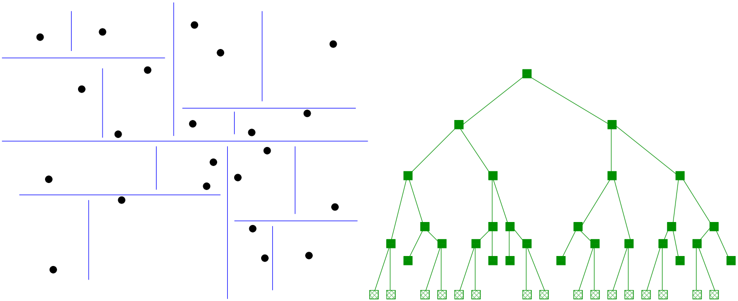
Kd-Trees in Pictures



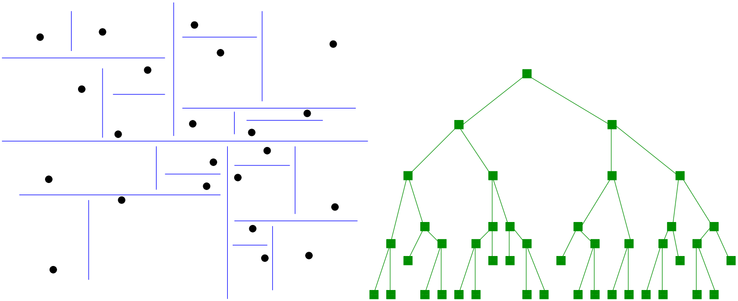
Kd-Trees in Pictures



Kd-Trees in Pictures



Kd-Trees in Pictures



Kd-Trees in Mathematics

- Split the points into two equal sets with a horizontal separator
- If the set has only one point, return a leaf with that point
- Else recursively construct a kd-tree on each set, but splitting vertically

Kd-Trees in Mathematics

- Split the points into two equal sets with a horizontal separator
- If the set has only one point, return a leaf with that point
- Else recursively construct a kd-tree on each set, but splitting vertically

More precisely, split horizontally at odd depth, vertically at even depth.

Kd-Trees in Equations

- $O(n)$ storage
- $O(n \log n)$ to build
- $O(\sqrt{n} + k)$ to query axis-parallel rectangles

Kd-Trees in Popular Culture

- Better worst-case behaviour than quadtrees
- Bentley, 1975 *(J. L. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM, 18:509–517, 1975)*
- kNN *(for small dimension)*

M. de Berg et al., *Computational Geometry: Algorithms and Applications*, second edition, chapter 5. Cf. also chapter end notes.

Outline

Range trees in Words

- Range queries again
- In $\mathbb{R}^1$, a balanced binary search tree

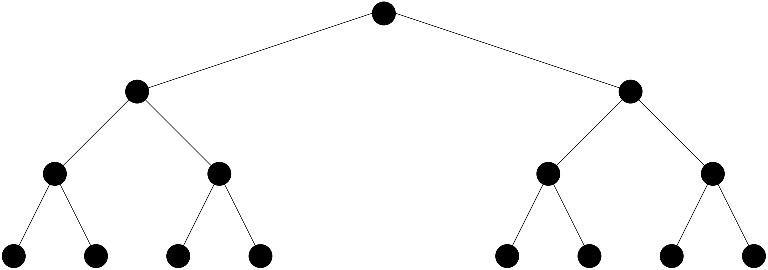
Range trees in Words

- Range queries again
- In $\mathbb{R}^1$, a balanced binary search tree
- In $\mathbb{R}^d$, split on dimension 1 with auxiliary range tree on remaining $d-1$ dimensions

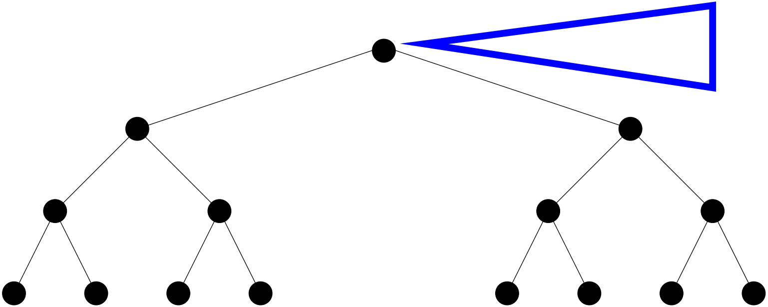
Range trees in Words

- Range queries again
- In $\mathbb{R}^1$, a balanced binary search tree
- In $\mathbb{R}^d$, split on dimension 1 with auxiliary range tree on remaining $d-1$ dimensions
- Faster but bigger

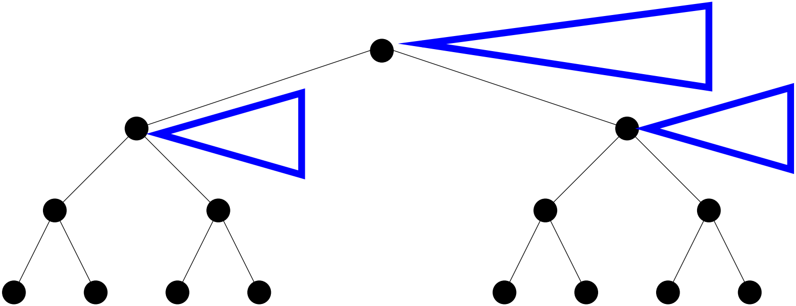
Range trees in Pictures



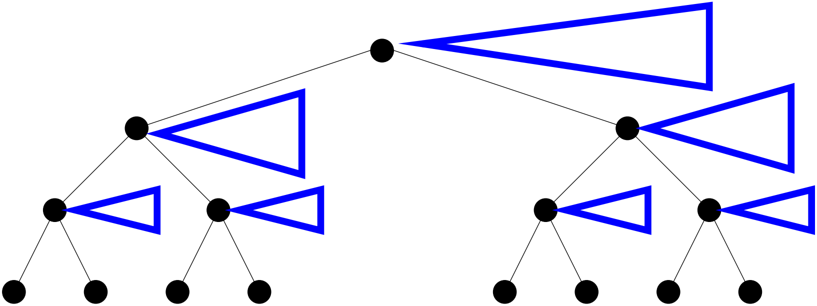
Range trees in Pictures



Range trees in Pictures



Range trees in Pictures



Range trees in Mathematics

- Build BST on first coordinate: $O(n \log n)$
- Recursively build range trees at each interior node on remaining $d - 1$ coordinates: ??

Range trees in Equations

Compared to kd-trees:

- Faster query times, $O(\log^d n + k)$
- Worse storage, $O(n \log^{d-1} n)$
- Worse construction, $O(n \log^{d-1} n)$

Range trees in Popular Culture

- Bentley, 1979 *J. L. Bentley, Decomposable searching problems, Information Processing Letters 8 (5): 244–251, 1979*
- Also: Lueker (1978); Lee, Wong (1980); Willard (1979)

M. de Berg et al., *Computational Geometry: Algorithms and Applications*, second edition, chapter 5. Cf. also chapter end notes.

Comparison

	Space	Time	Query
quadtree	$O\left(n \log\left(\frac{s}{c}\right)\right)$	$O\left(n \log\left(\frac{s}{c}\right)\right)$	$O\left(m \log\left(\frac{s}{c}\right)\right)$
kd-tree	$O(n)$	$O(n \log n)$	$O(\sqrt{n} + k)$
range-tree	$O(n \log^{d-1} n)$	$O(n \log^{d-1} n)$	$O(\log^{d} n + k)$

Outline

$\mathbb{R}^1$

- BST

$\mathbb{R}^d$

- What could go wrong?
- Quadtrees
- kd-trees
- Range trees
- More

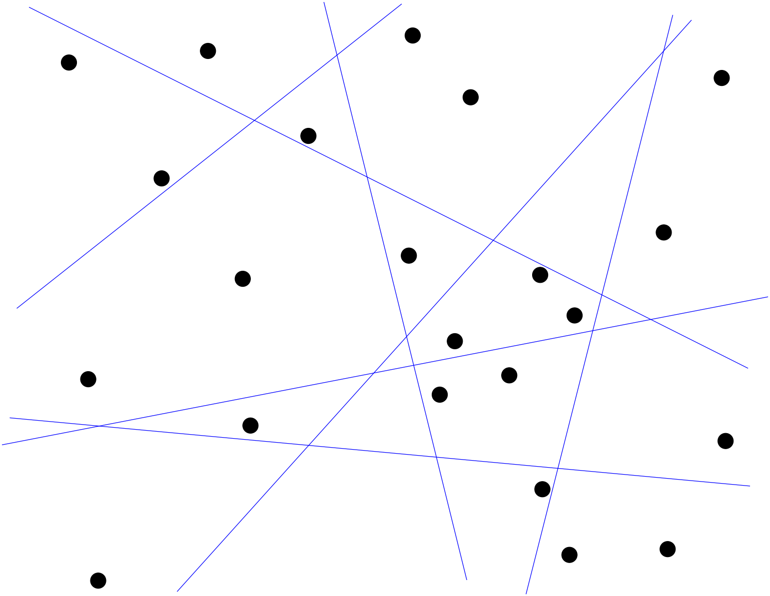
More. . .

- Voronoi diagrams
- database range searches
- kNN *(Use kd-trees for small d)*
- aNN (locality sensitive hashing)
- What if we want to insert or delete points?

aNN and LSH

- N points
- h hyperplanes, $h \ll N$
- planes are oriented
- Hamming distance, e.g.

aNN and LSH

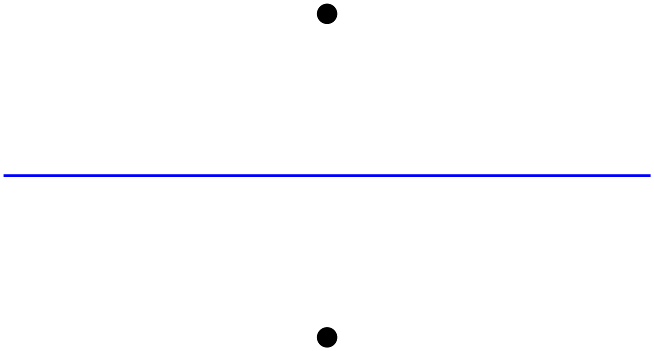


Voronoi Diagrams in Words

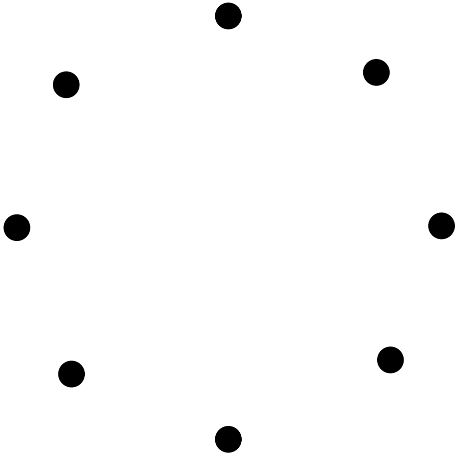
- 1-nearest-neighbour
- Grass fire function

Voronoi Diagrams in Pictures

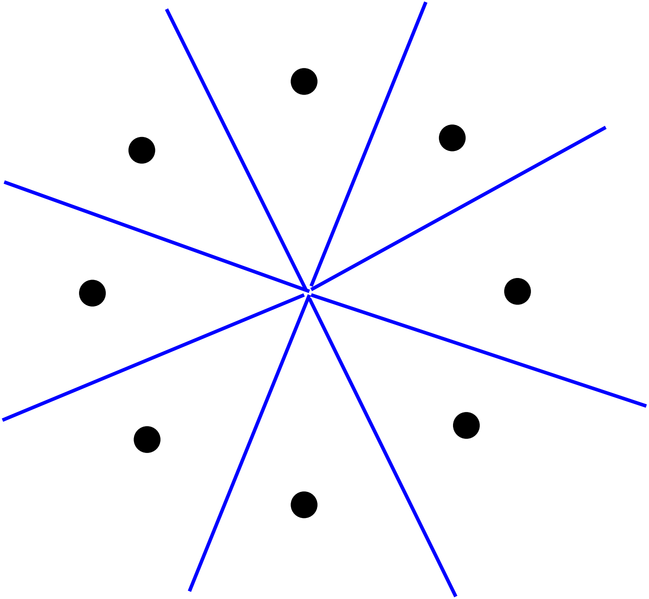
Voronoi Diagrams in Pictures



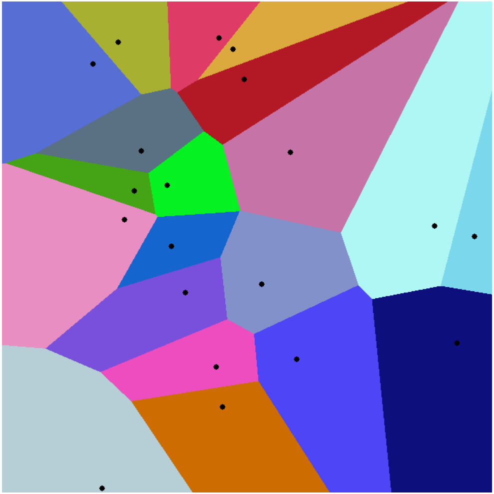
Voronoi Diagrams in Pictures



Voronoi Diagrams in Pictures

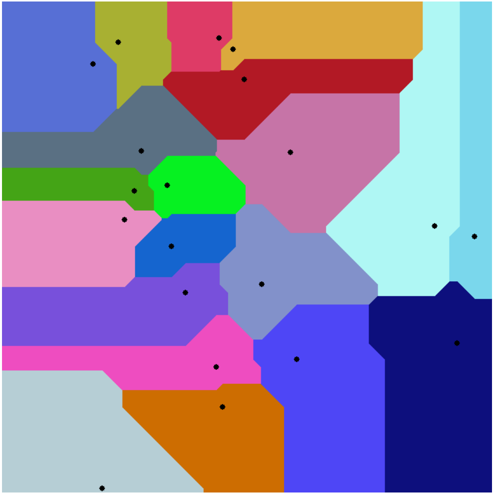


Voronoi Diagrams in Pictures



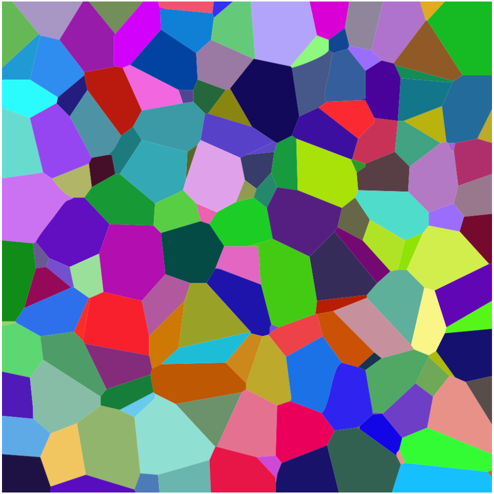
https://en.wikipedia.org/wiki/File:Euclidean_Voronoi_Diagram.png

Voronoi Diagrams in Pictures



https://en.wikipedia.org/wiki/File:Manhattan_Voronoi_Diagram.png

Voronoi Diagrams in Pictures



A slice of a 3D voronoi diagram on random points.

`https://en.wikipedia.org/wiki/File:Coloured_Voronoi_3D_slice.svg.png`

Voronoi Diagrams in Mathematics

- Sweep line algorithm
- We don't have enough time today...

Voronoi Diagrams in Equations

- $O(n)$ to store
- $O(n \log n)$ to construct

Voronoi Diagrams in Popular Culture

- Shamos and Hoey, 1975 *(M.I. Shamos and D. Hoey, Closest-point problems, in Proc. 16th Annual IEEE Sympos. Found. Comput. Sci., pp 151–162, 1975)*
- Delaunay triangulations *(2D)*

M. de Berg et al., *Computational Geometry: Algorithms and Applications*, second edition, chapter 7. Cf. also chapter end notes.

Summary

- Trade-offs
- Every problem is different
- Dimension matters
- This is all quite simplified

Questions