TU Graz

# Recent Algorithmic Advances in Topological Data Analysis

**Michael Kerber**

Gudhi workshop, Porquerolles, France, Oct 19, 2016

# Computational Topology@TU Graz

Arnur Nigmetov

Hannah Schreiber

Aruni Choudhary
(MPI Saarbrücken)

# Our mission

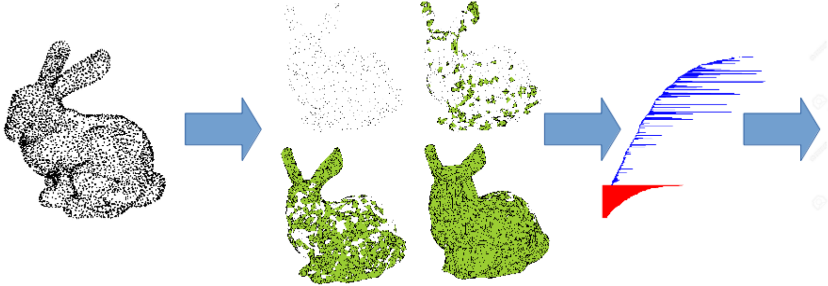
"... to boldly compute what no topologists has computed before."

# Our mission

"… to boldly
compute what no
topologists has
computed before."

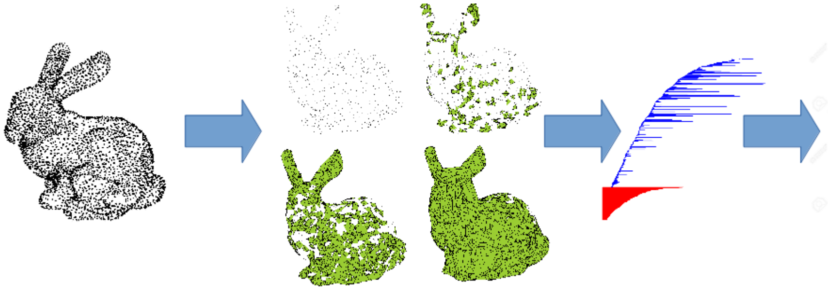
Algorithmic foundations, implementations, and software in computational topology and geometry.

# The algorithmic pipeline



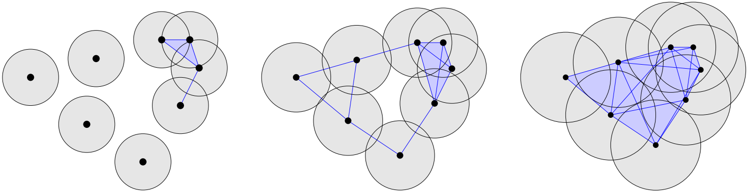
1. Turn input into multi-scale representation
2. Compute topological invariants
3. Draw conclusions about the input

# The algorithmic pipeline



1. **Turn input into multi-scale representation**
2. Compute topological invariants
3. **Draw conclusions about the input**

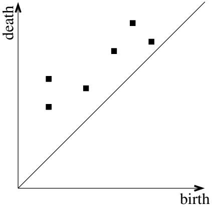
# Čech filtrations



- Nested sequence of simplicial complexes
- Important in topological data analysis
- Vietoris-Rips complexes: Closely related
- **Problem**: Size of $k$-skeleton is $\binom{n}{k+1} = O(n^{k+1})$

# Topological approximation

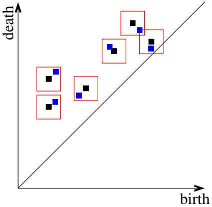
Persistence diagram of the Čech filtration:




**Question:** Can we find a small filtration whose persistence diagram is provably close to the Čech diagram?

# Topological approximation

Persistence diagram of the Čech filtration:




**Question:** Can we find a small filtration whose persistence diagram is provably close to the Čech diagram?

# Previous work

- Sparse Rips complex [Sheehy 2012]
- $(1+\varepsilon)$-approximation of size

$$n \cdot \left(\frac{1}{\varepsilon}\right)^{O(\Delta k)}$$

  with $\Delta$ the doubling dimension of the point set
- Various related approaches [Dey, Fan, Wang 2012] [K., Sharathkumar 2013] [Botnan, Spreemann 2015] [Buchet et al. 2015] [Cavanna, Jahanseir, Sheehy 2015]

# Our contributions [Choudhary,K.,Raghvendra, SoCG 2016]

- $6(d+1)$-approximation of size

$$n \cdot 2^{O(d \log k)}$$

per scale for $n$ points in $\mathbb{R}^d$.

# Our contributions [Choudhary,K.,Raghvendra, SoCG 2016]

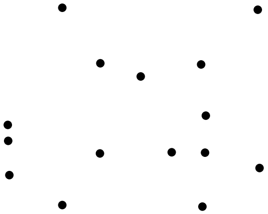
- $6(d+1)$-approximation of size

$$n \cdot 2^{O(d \log k)}$$

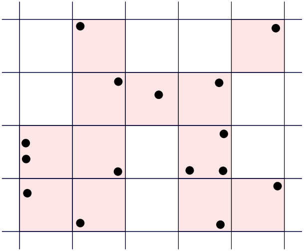
per scale for $n$ points in $\mathbb{R}^d$.

- Combined with dimension reduction: $O(\log^{3/2} n)$-approximation of size $n^{O(1)}$.

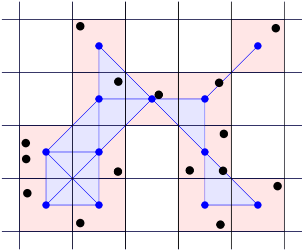
# Approximation by lattices I



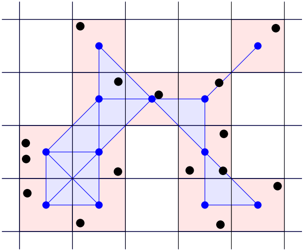
# Approximation by lattices I



# Approximation by lattices I



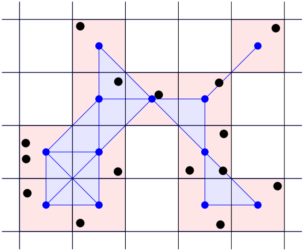
# Approximation by lattices I



- Diameter of a cell: $\alpha \cdot \sqrt{d}$
- Two non-adjacent cells are at least $\alpha$ apart

$\Rightarrow \sqrt{d}$-approximation!

# Approximation by lattices I

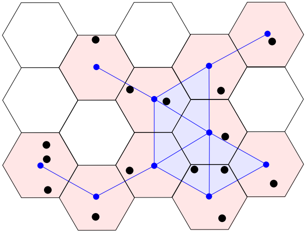


- Diameter of a cell: $\alpha \cdot \sqrt{d}$
- Two non-adjacent cells are at least $\alpha$ apart

$\Rightarrow \sqrt{d}$-approximation!

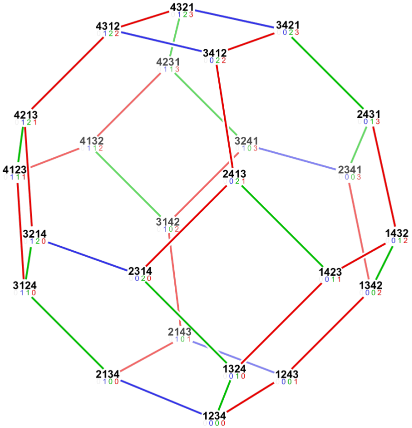
But highly degenerate: $2^d$ cells intersect in a point (leads to size $n \cdot 2^{O(dk)}$)

# Approximation by lattices II



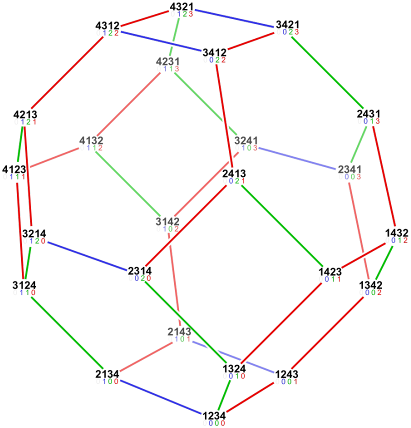
- Hexagonal grid
- How to generalize in higher dimensions?

# The permutahedron



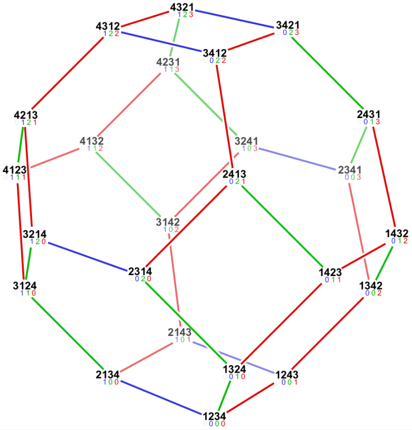

- Voronoi region of $A_d^*$-lattice

# The permutahedron



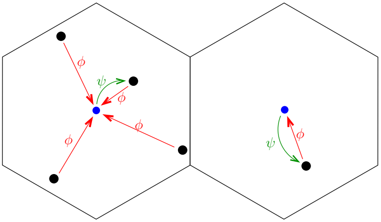

- Voronoi region of $A_d^*$-lattice
- Diameter: $O(\alpha \cdot \sqrt{d})$
- Lemma: Non-intersecting cells are at least $\alpha \cdot \sqrt{2}/(d+1)$ apart.

# The permutahedron




- Voronoi region of $A_d^*$-lattice
- Diameter: $O(\alpha \cdot \sqrt{d})$
- Lemma: Non-intersecting cells are at least $\alpha \cdot \sqrt{2}/(d+1)$ apart.
- Size of the dual $k$-skeleton: $n2^{O(d \log k)}$

# Interleaving




$$
\begin{tikzcd}
 & \cdots \arrow[r] & \mathcal{R}_{\beta 2(d+1)} \arrow[rr, "g"] \arrow[dr, "\phi"] & & \mathcal{R}_{\beta 8(d+1)^3} \arrow[r] & \cdots \\
\cdots \arrow[r] & X_\beta \arrow[ur, "\psi"] \arrow[rr, "\theta"] & & X_{\beta 4(d+1)^2} \arrow[ur, "\psi"] \arrow[r] & \cdots &
\end{tikzcd}
$$

# The Johnson-Lindenstrauss Lemma

For a point set $S \subset \mathbb{R}^d$ of $n$ points and $0 < \varepsilon < 1$, there is a map

$$f : \mathbb{R}^d \to \mathbb{R}^m$$

with $m = O(\frac{\log n}{\varepsilon^2})$ such that for any two points $s, t \in S$:

$$(1 - \varepsilon)\|s - t\| \leq \|f(s) - f(t)\| \leq (1 + \varepsilon)\|s - t\|.$$

# The Johnson-Lindenstrauss Lemma

For a point set $S \subset \mathbb{R}^d$ of $n$ points and $0 < \varepsilon < 1$, there is a map

$$f : \mathbb{R}^d \to \mathbb{R}^m$$

with $m = O(\frac{\log n}{\varepsilon^2})$ such that for any two points $s, t \in S$:

$$(1-\varepsilon)\|s-t\| \leq \|f(s)-f(t)\| \leq (1+\varepsilon)\|s-t\|.$$

Moreover, a random (scaled) projection from $\mathbb{R}^d$ to $\mathbb{R}^m$ has that property with a probability of at least $\frac{1}{2}$.

# Dimension reduction

- Size per scale $n \cdot 2^{O(d \log k)}$
- [Johnson, Lindenstrauss 1984]: $d \approx \log n$ (constant distortion) $\Rightarrow$ size $n^{O(\log k)}$, total distortion $O(\log n)$

# Dimension reduction

- Size per scale $n \cdot 2^{O(d \log k)}$
- [Johnson, Lindenstrauss 1984]: $d \approx \log n$ (constant distortion)
  $\Rightarrow$ size $n^{O(\log k)}$, total distortion $O(\log n)$
- [Matoušek 1990]: $d \approx \frac{\log n}{\log \log n}$ (($\log n$)-distortion)
  $\Rightarrow$ size $n^{O(1)}$, total distortion $O(\log^2 n)$

# Dimension reduction

- Size per scale $n \cdot 2^{O(d \log k)}$
- [Johnson, Lindenstrauss 1984]: $d \approx \log n$ (constant distortion) $\Rightarrow$ size $n^{O(\log k)}$, total distortion $O(\log n)$
- [Matoušek 1990]: $d \approx \frac{\log n}{\log \log n}$ (($\log n$)-distortion) $\Rightarrow$ size $n^{O(1)}$, total distortion $O(\log^2 n)$
- [Bourgain 1985]: General metric space: Embed to $O(\log^2 n)$ dimensions with distortion $O(\log n)$ $\Rightarrow$size $n^{O(1)}$ and total distortion $O(\log^3 n)$

# Our contributions [Choudhary,K.,Raghvendra, SoCG 2016]

- $6(d+1)$-approximation of size

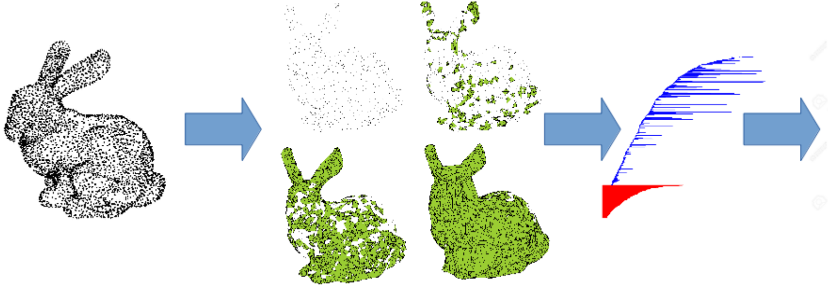
$$n \cdot 2^{O(d \log k)}$$

per scale for $n$ points in $\mathbb{R}^d$.

- Combined with dimension reduction: $O(\log^{3/2} n)$-approximation of size $n^{O(1)}$.

- Lower bound: Any $(1+\delta)$-approximation scheme has to be of size $n^{\Omega(\log \log n)}$ if $\delta < \frac{1}{96 \log^{1.001} n}$.

# The algorithmic pipeline



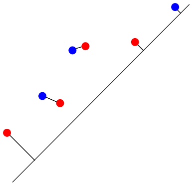
1. **Turn input into multi-scale representation**
2. Compute topological invariants
3. **Draw conclusions about the input**

# Distances between persistence diagrams


| | $A_1$ | $A_2$ | $\ldots$ | $A_{n-1}$ | $A_n$ |
|---|---|---|---|---|---|
| $A_1$ | | | | | |
| $A_2$ | | | | | |
| $\vdots$ | | | $d(A_i, A_j)$ | | |
| $A_{n-1}$ | | | | | |
| $A_n$ | | | | | |

- $n$ diagrams $\Rightarrow \binom{n}{2}$ distances
- Often the computational bottleneck

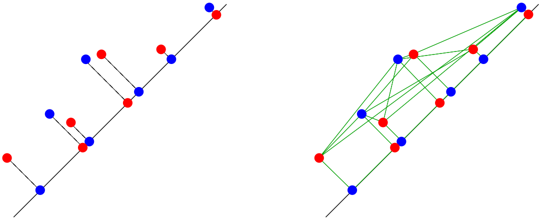
# Distance measures



- One-to-one pairing of points
- Every point must be paired
- Pairing with diagonal allowed

- Cost of a pair $(p, q)$: $\|p - q\|_\infty$
- **Bottleneck distance**: Minimize maximal cost
- 1-**Wasserstein distance**: Minimize $\Sigma$ of the costs
- Stability [Cohen-Steiner et al. 2007]
- Other distances

# From diagram distance to graph matching



- Weighted complete bipartite graph $G$
- Weight of an edge: $L_\infty$ distance of the points
- EXCEPT: weight is zero if both points are on diagonal
- Use graph matching algorithm (Hopcroft-Karp, Hungarian, ...)

# Does geometry help?

- *G* is “almost” metric (modulo diagonal)
- Asymptotically faster algorithms are known for this case
- [Efrat, Itai, Katz: Geometry Helps in Bottleneck Matching... 2001]
- [Vaidya: Geometry Helps in Matchings. 1989] (opt. assignment)
- Adaption to persistence diagrams straight-forward
  [Folklore? Mentioned in Edelsbrunner, Harer 2010]

# Does geometry help?

- *G* is “almost” metric (modulo diagonal)
- Asymptotically faster algorithms are known for this case
- [Efrat, Itai, Katz: Geometry Helps in Bottleneck Matching... 2001]
- [Vaidya: Geometry Helps in Matchings. 1989] (opt. assignment)
- Adaption to persistence diagrams straight-forward
  [Folklore? Mentioned in Edelsbrunner, Harer 2010]

**Answer**

**Yes**, in theory

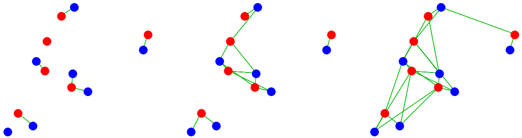
# Does geometry help?

## Our contribution

**Yes**, also in practice! [K., Morozov, Nigmetov, ALENEX 2016]

- We compare geometric and non-geometric implementations of bottleneck matchings and optimal assignment (for $\mathbb{R}^2$)
- We show experimentally that geometry improves performance
- We outperform Dionysus, the only publically available software for distances of persistence diagrams
- Our code is freely available: `https://bitbucket.org/grey_narn/hera`

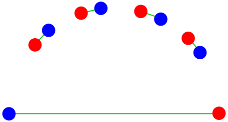
# The bottleneck case



- Let $G[\alpha]$ be the graph $G$ with all edges of weight $> \alpha$ deleted
- Observation: If $G[\alpha]$ has a perfect matching, the bottleneck distance is at most $\alpha$.

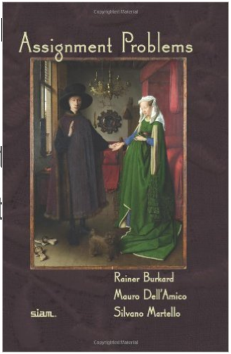
# The Wasserstein case

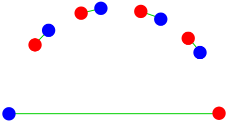
- Assignment problem: Find perfect matching with minimal sum of costs.
- Discrete optimal transport
- Hungarian algorithm



# The Wasserstein case

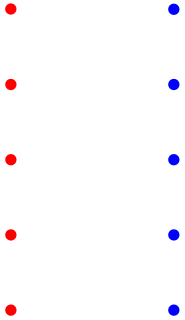
- Assignment probl
perfect matching
sum of costs.
- Discrete optimal t
- Hungarian algorit



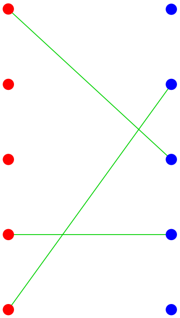



# The auction algorithm [Bertsekas 1988]

- *n* bidders (left), *n* objects (right)

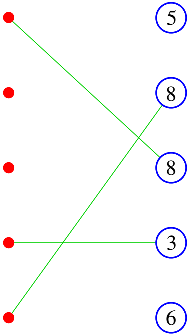


# The auction algorithm [Bertsekas 1988]



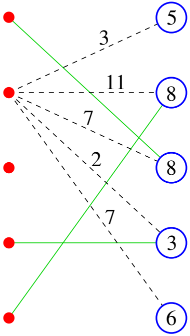
- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching

# The auction algorithm [Bertsekas 1988]



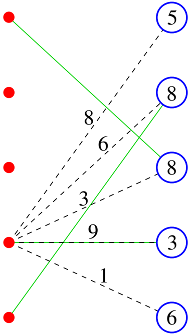

- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching
- Objects have (global) prices

# The auction algorithm [Bertsekas 1988]



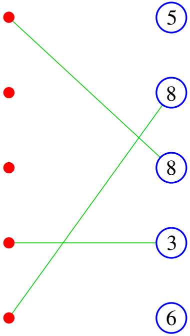

- *n* bidders (left), *n* objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

# The auction algorithm [Bertsekas 1988]




- *n* bidders (left), *n* objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

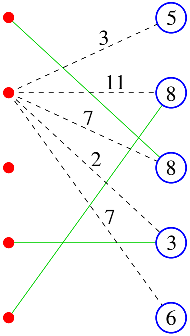
# The auction algorithm [Bertsekas 1988]




- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

Repeat:

# The auction algorithm [Bertsekas 1988]

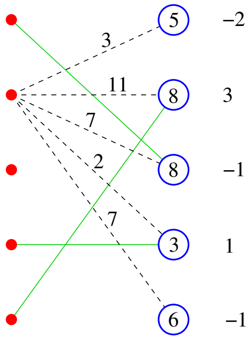



- *n* bidders (left), *n* objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

Repeat:

- Pick an unassigned bidder

# The auction algorithm [Bertsekas 1988]

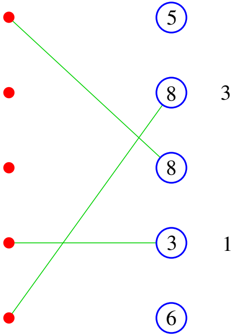



- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

Repeat:

- Pick an unassigned bidder
- Value of object: appreciation - price

# The auction algorithm [Bertsekas 1988]

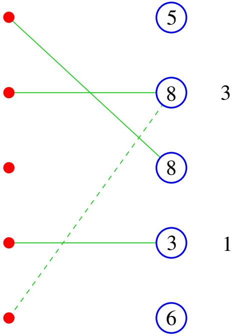



- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

Repeat:

- Pick an unassigned bidder
- Value of object: appreciation - price
- Pick best and 2nd-best valued objects

# The auction algorithm [Bertsekas 1988]

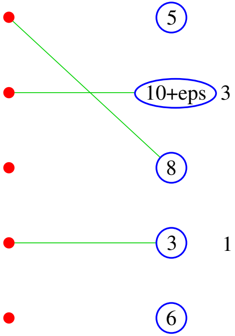



- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

Repeat:

- Pick an unassigned bidder
- Value of object: appreciation - price
- Pick best and 2nd-best valued objects
- Assign bidder to best valued object

# The auction algorithm [Bertsekas 1988]




- $n$ bidders (left), $n$ objects (right)
- Maintain (partial) matching
- Objects have (global) prices
- Bidders have (individual) appreciations

Repeat:

- Pick an unassigned bidder
- Value of object: appreciation - price
- Pick best and 2nd-best valued objects
- Assign bidder to best valued object
- Increase price by difference of values, plus $\varepsilon > 0$

# Why auction?

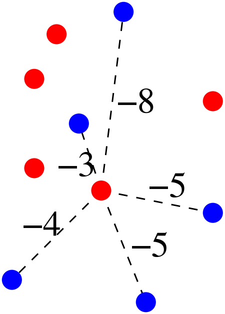
**Theorem** [Bertsekas 1988]

Let opt denote the cost of the optimal assignment, and $d$ the cost returned by the auction. Then

$$\text{opt} - n\varepsilon \le d \le \text{opt}$$

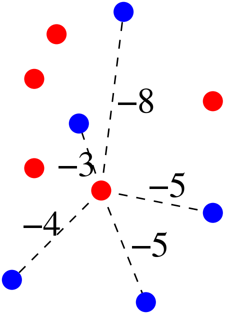
- Large $\varepsilon$: Fast, rough approximation
- Small $\varepsilon$: Slow, accurate approximation
- $\varepsilon$-scaling [Bertsekas, Castanon 1991]
- Remark: Getting exact result possible, but very slow

# How geometry helps




- Appreciation = -distance to object
- Crucial query: Find the best and second best object for an unassigned bidder.

# How geometry helps

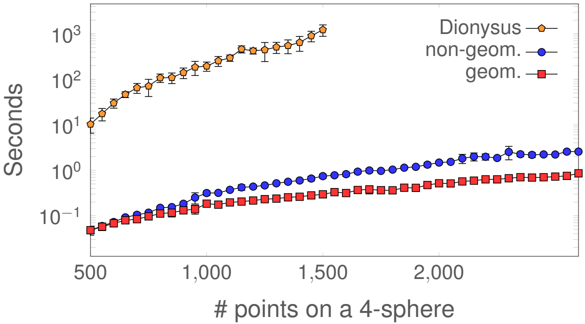



- Appreciation = -distance to object
- Crucial query: Find the best and second best object for an unassigned bidder.

## Our approach

- k-d-tree with weight per node
- Weight=minimal price
- Prune search in subtrees if better candidates are known

# Experimental comparison

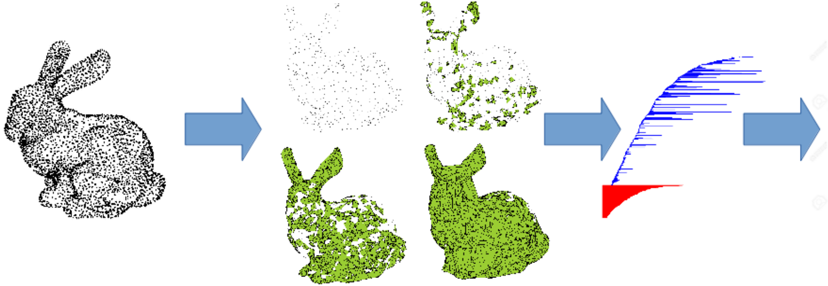



- Linear space (geometric) vs quadratic space (non-geometric)
- Exact distance (Dionysus) vs relative 1%-approximation

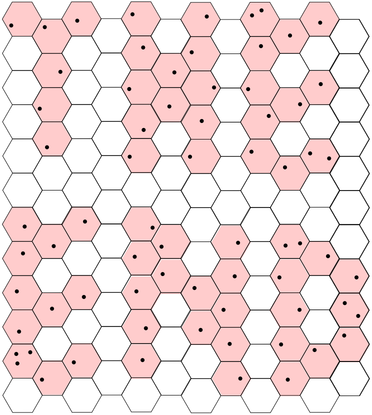
# The Hera library

- URL: `https://bitbucket.org/grey_narn/hera`
- Code for bottleneck (LGPL) and Wasserstein (BSD)
- Supports $q$-Wasserstein distance and different choice of inner metric (instead of $L_\infty$)
- Download it! Use it! Tell us your experience!

# The algorithmic pipeline



1. Turn input into multi-scale representation
2. Compute topological invariants
3. Draw conclusions about the input




THANK YOU