Hausdorff Distance under Translation for Points, Disks, and Balls

Yusu Wang

Joint work with

P. K. Agarwal, S. Har-Peled, and M. Sharir

BioGeometry

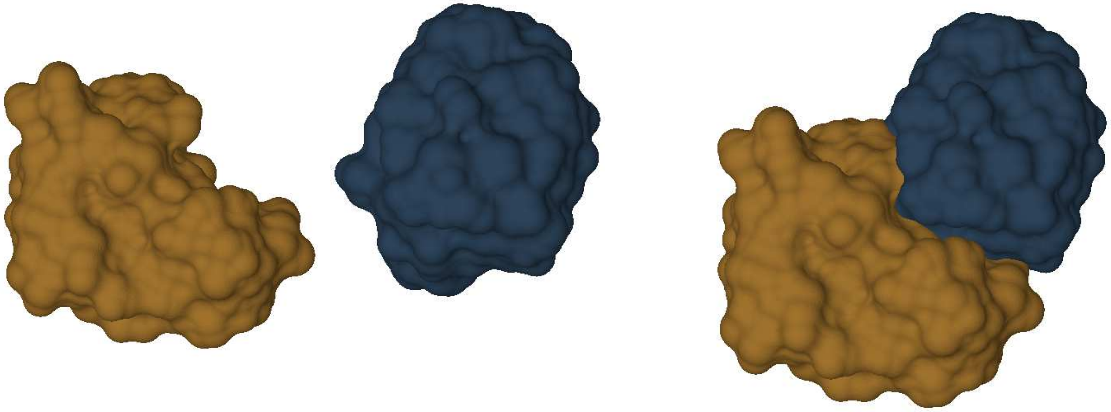
Center for Geometric and Biological Computing

Duke University

Introduction

Surface Matching: Find the best *transformation* s.t. two *surfaces* are most *similar*.

Motivation: Computer vision, CAD, graphics, robotics and molecular biology.

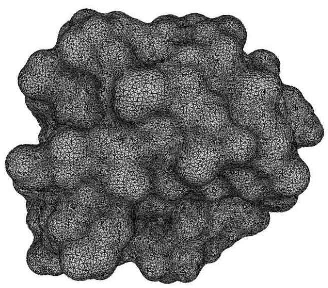


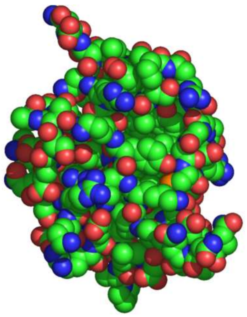
Molecular Shape Matching

- ☆ Proteins w/ similar shapes likely have similar functionalities
- ☆ Protein-protein or protein-ligand docking

Representation of surfaces:

- ☆ points, union of balls, weighted points





Transformation space:

- ☆ translational space in $\mathbb{R}^2$ or $\mathbb{R}^3$

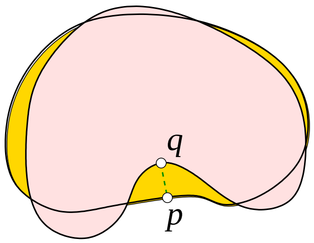
Overview

Similarity measure: variants of Hausdorff distance.

- ☆ Hausdorff distance between unions of balls
- ☆ Collision-free Hausdorff between sets of weighted points
- ☆ Average Hausdorff between sets of points (approximate)
- ☆ Other approximation algorithms

Hausdorff between unions

- $\mathcal{A} = \{A_1, \ldots, A_m\}$, $A_i = D(a_i, r_i)$
- $\mathcal{B} = \{B_1, \ldots, B_n\}$, $B_j = D(b_j, \rho_j)$
- $U_A = \bigcup_i A_i, U_B = \bigcup_j B_j$.



- **Directional Hausdorff:**

$$\mathrm{h}_U(\mathcal{A}, \mathcal{B}) = \max_{p \in U_A} \min_{q \in U_B} d(p, q),$$

- **Hausdorff:**

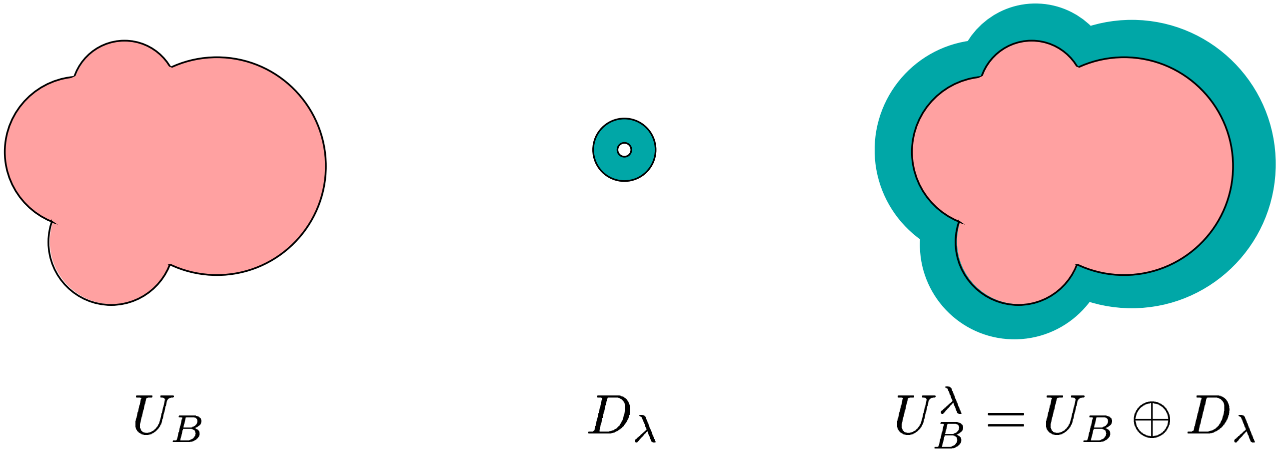
$$\mathrm{H}_U(\mathcal{A}, \mathcal{B}) = \max\{\mathrm{h}_U(\mathcal{A}, \mathcal{B}), \mathrm{h}_U(\mathcal{B}, \mathcal{A})\},$$

- **Goal:**

$$\sigma_U(\mathcal{A}, \mathcal{B}) = \inf_{t \in \mathbb{R}^2} \mathrm{H}_U(\mathcal{A} + t, \mathcal{B}).$$

Framework

- ☆ **Decision problem:** Given $\lambda \geq 0$, $\exists t \in \mathbb{R}^2$ s.t. $\mathrm{h}_U(\mathcal{A} + t, \mathcal{B}) \leq \lambda$?

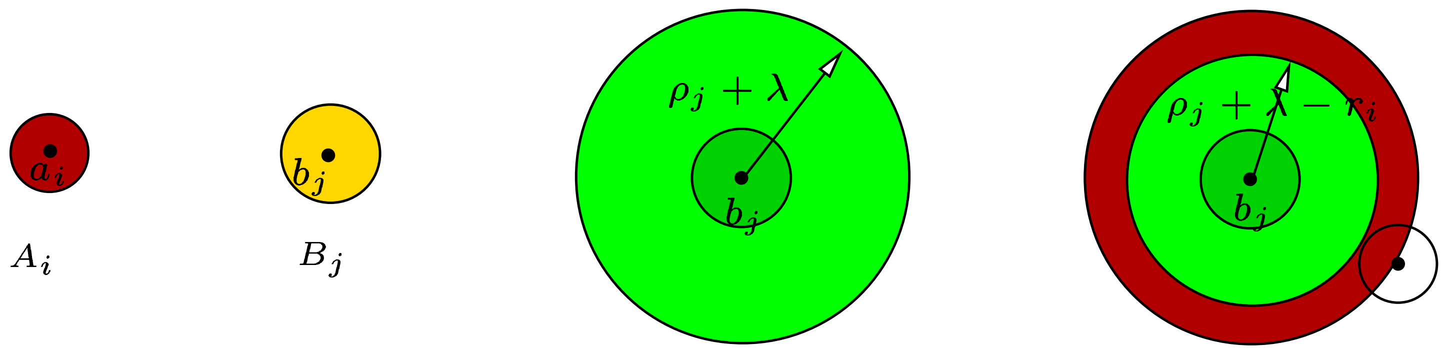


- ☆ $V_i \subseteq \mathbb{R}^d$: placements t's of A_i s.t. $\mathrm{h}_U(A_i + t, \mathcal{B}) \leq \lambda$
- ☆ $V(\mathcal{A}, \mathcal{B}) = \bigcap_{1 \leq i \leq m} V_i$
- ☆ **Goal:** Is $V(\mathcal{A}, \mathcal{B})$ empty?

BioGeometry

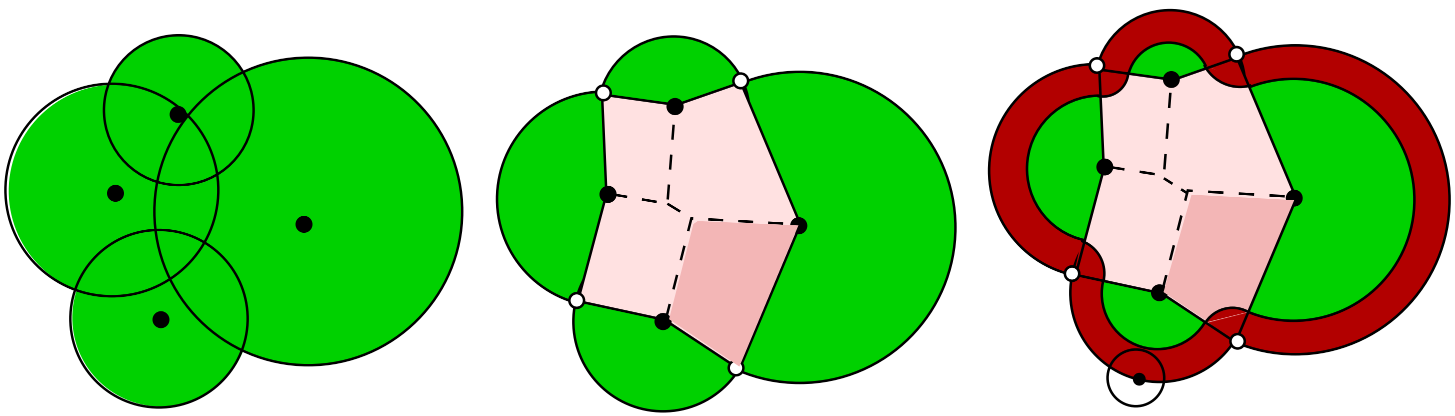
Structure of V_i

- ☆ One pair of disks $A_i = D(a_i, r_i)$ and $B_j = D(b_j, \rho_j)$
- ☆ Placements s.t. $\mathrm{H}_U(A_i + t, B_j) \le \lambda$
- ☆ $D_{ij} = D(b_j - a_i, \rho_j + \lambda - r_i)$



Structure of V_i

- ☆ Same procedure A_i and U_B
- ☆ Q_i: vertex set for U_B^λ, Γ_i: arc set for U_B^λ



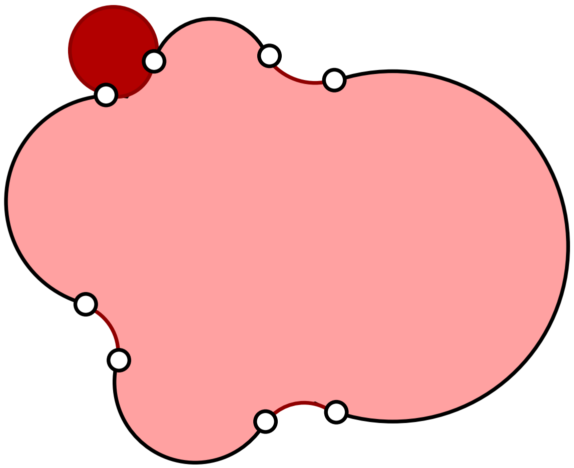
- ☆ Complexity of V_i : $O(n)$ in $\mathbb{R}^2$

Complexity of $V(\mathcal{A}, \mathcal{B})$

- ☆ $V(\mathcal{A}, \mathcal{B}) = \bigcap_i V_i$, for $1 \leq i \leq m$.

Lemma. Complexity of $V(\mathcal{A}, \mathcal{B})$: $O(m^2 n)$ in $\mathbb{R}^2$.

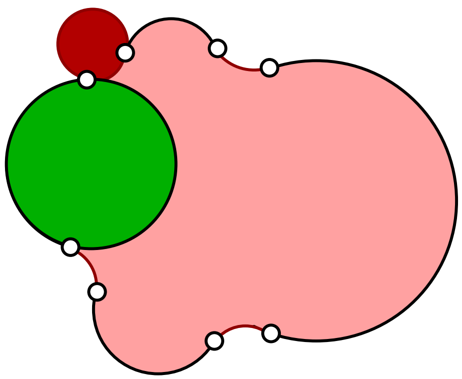
Proof: A vertex can be from:



- ☆ Some V_i — $O(nm)$;
- ☆ Intersection of an arc from V_i and one from V_k
  - $O(n)$ vertices for one pair
  - $O(m^2)$ pairs of (V_i, V_k)
  - $O(m^2 n)$ overall

$\partial V_i \cap \partial V_k$

For ∂V_i: consider two types of disks:



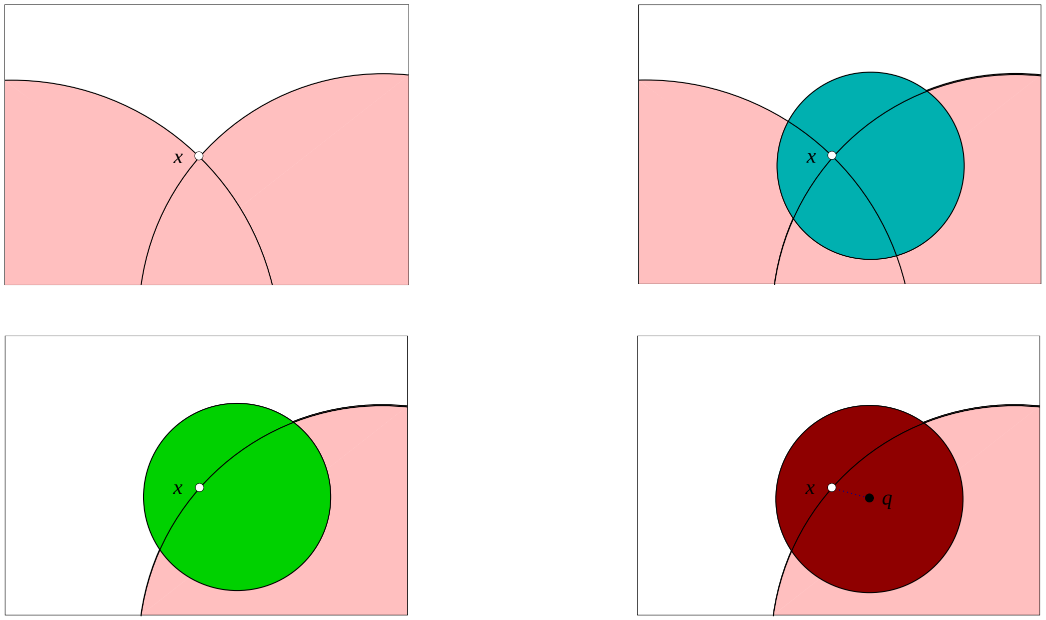
- ☆ convex arcs bounded by:

$$D_i = \{D_{ij}, 1 \leq j \leq n\}$$

- ☆ concave arcs bounded by:

$$\Delta_i = \{D(q, r_i), q \in Q_i\}$$

Claim. $\partial V_i \cap \partial V_k \subseteq \partial \bigcup (D_i \cup \Delta_i \cup D_k \cup \Delta_k)$.



Algorithm

Lemma. $V(\mathcal{A}, \mathcal{B})$ can be computed in time $O(m^2 n \log(n+m))$ in $\mathbb{R}^2$.

- ☆ Divide and conquer
- ☆ Sweep-line approach to merge

Theorem. $\sigma_U(\mathcal{A}, \mathcal{B})$ can be computed in $O(mn(n+m)\log^3(n+m))$.

- ☆ Parametric search technique.

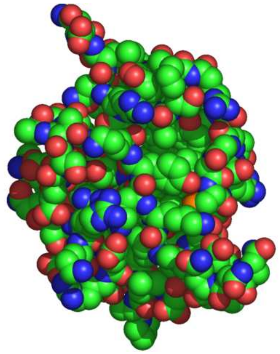
Computing $V(\mathcal{A}, \mathcal{B})$ in $\mathbb{R}^3$

Bad news: $\tilde{O}(n^7)$ in $\mathbb{R}3$!!

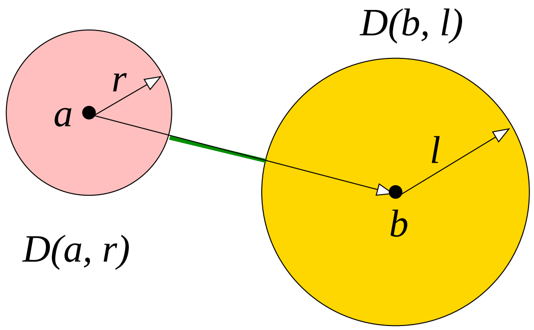
- Open problem: Complexity of V_i : $O(n^4)$
  - what is complexity of medial axis for union of balls

Remark: $\tilde{O}(nm(n+m)^2)$ under assumptions for molecules:

- Atoms of similar (constant) size
- Atoms are constant distance away
- λ is some constant



Weighted Points



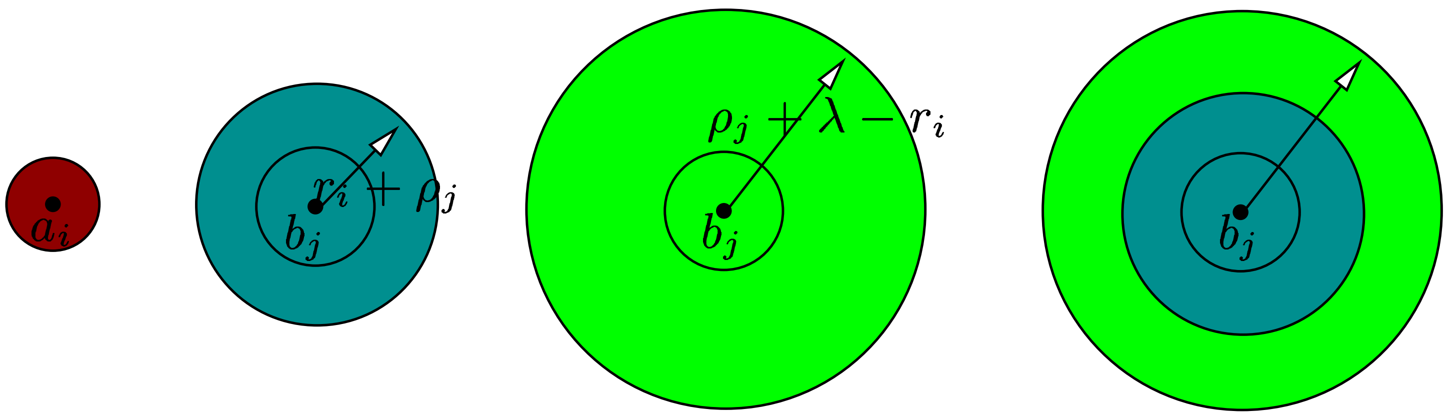
- $d(A_i, B_j) = d(a_i, b_j) - r_i - \rho_j$
- $\mathcal{F}$: the set of free placements of $\mathcal{A}$
  - $x \in \mathbb{R}^d$ *free*: $\mathcal{A} + x$ avoids interior of $\mathcal{B}$

Goal: Optimal collision-free Hausdorff under translations:

$$\mathrm{H}(\mathcal{A}, \mathcal{B}; \mathcal{F}) = \min_{x \in \mathcal{F}} \mathrm{H}(\mathcal{A} + x, \mathcal{B}).$$

Framework

- F_i: set of placements of A_i that are not free
- Similar framework as before
  - distance condition (V_i) and collision-free condition (F_i)
  - $V(\mathcal{A}, \mathcal{B}) = \bigcap (V_i \setminus F_i)$
- Take $A_i = (a_i, r_i)$ and $B_j = (b_j, \rho_j)$



Results

☆ Follow similar framework as before:

Theorem. $\mathrm{H}(\mathcal{A}, \mathcal{B}; \mathcal{F})$ can be computed

(i) $\mathbb{R}^2$: in time $O(mn(n+m)\log^3(n+m))$;

(ii) $\mathbb{R}^3$: in time $O(m^2n^2(m+n)\log^3(n+m))$.

Remark:

☆ Bounds approximately same as Hausdorff between point sets

Partial Matching

Motivation of $d(A_i, B_j)$: docking problem, partial matching

Partial matching: Given $\lambda \geq 0$, find $x \in \mathcal{F}$ s.t.

$$|\{A_i \mid \mathrm{H}(A_i + x, \mathcal{B}) \leq \lambda\}|$$

is maximized.

Theorem. Such a translation x can be computed

(i) $\mathbb{R}^2$: in time $O(m^2 n \log(n+m))$;

(ii) $\mathbb{R}^3$: in time $O(m^3 n^2 \log(n+m))$.

Open problem:

☆ How to approximate partial matching efficiently?

Open Problems

- Rigid motions
- Efficient approximation algorithms
  - approximate partial matching under rigid motion

—— THE END! ——

THANKS !

Related Work

Point sets:

- ☆ Exact matching (rigid motion): $O(n \log n)$ in $\mathbb{R}^2$, $O(n^{d-2} \log n)$ in $d \geq 3$
- ☆ Bottleneck (in $\mathbb{R}^2$): $O(n^{1.5} \log n)$ for fixed point sets, $O(n^5 \log n)$ translations, $O(n^8)$ rigid motion
- ☆ Hausdorff (translations, L_2): $\tilde{O}(n^3)$ in $\mathbb{R}^2$, $\tilde{O}(n^{\lceil 3d/2 \rceil + 1})$ in $\mathbb{R}^d$
- ☆ Hausdorff (rigid motions): $\tilde{O}(n^6)$ in $\mathbb{R}^2$

Other objects:

- ☆ Line segments (in $\mathbb{R}^2$): $O(n^2)$ for fixed sets, $\tilde{O}(n^4)$ translations, $\tilde{O}(n^6)$ rigid motions