Farthest Color Voronoi diagrams: conditions and algorithms

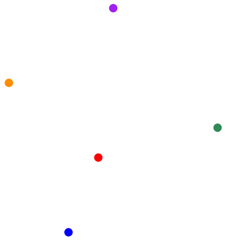
Ioannis Mantas[1] Evanthia Papadopoulou[1]
Vera Sacristán[2] Rodrigo I. Silveira[2]

[1] Università della Svizzera italiana, Lugano, Switzerland
[2] Universitat Politècnica de Catalunya, Barcelona, Spain

6/1/2021 - São Paulo, Brazil - LATIN 2020

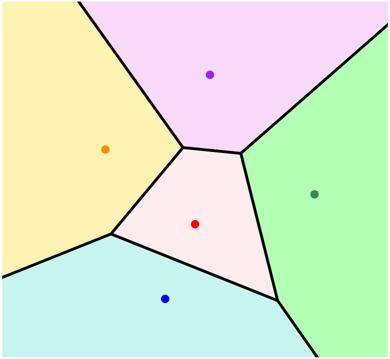
Point Voronoi Diagrams

- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



Point Voronoi Diagrams

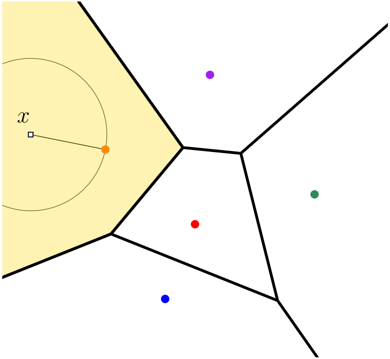
- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



- (Nearest point) **Voronoi diagram**

Point Voronoi Diagrams

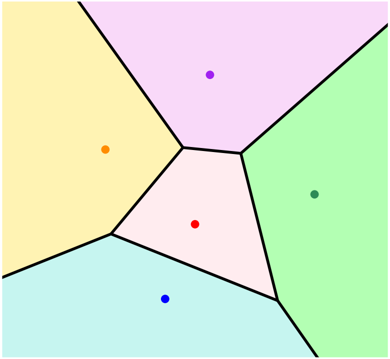
- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



- (Nearest point) **Voronoi diagram**
- (Nearest) **Voronoi region** of site s: $\{x \in \mathbb{R}^2 \mid d(x,s) < d(x,t)\ \forall t \in \mathcal{P}\}$

Point Voronoi Diagrams

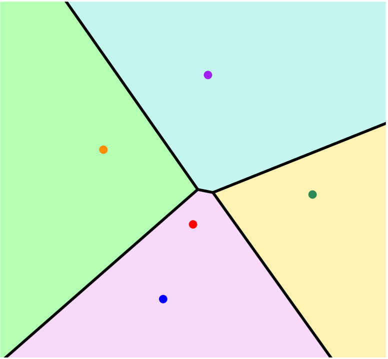
- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



- (Nearest point) **Voronoi diagram**
- (Nearest) **Voronoi region** of site s: $\{x \in \mathbb{R}^2 \mid d(x,s) < d(x,t)\ \forall t \in \mathcal{P}\}$

Point Voronoi Diagrams

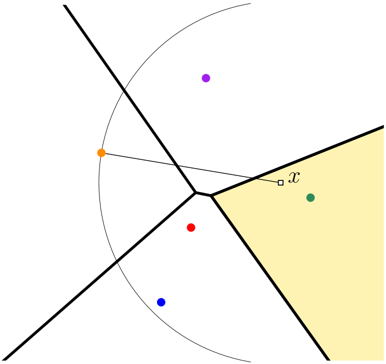
- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



- **Farthest** (point) **Voronoi diagram**

Point Voronoi Diagrams

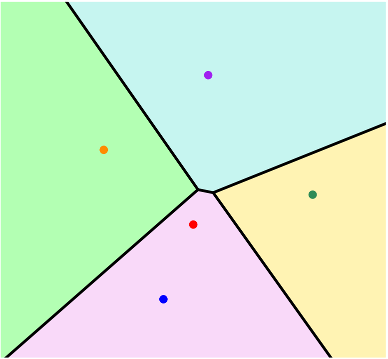
- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



- **Farthest** (point) **Voronoi diagram**
- **Farthest Voronoi region** of site s: $\{x \in \mathbb{R}^2 \mid d(x,s) > d(x,t)\ \forall t \in \mathcal{P}\}$

Point Voronoi Diagrams

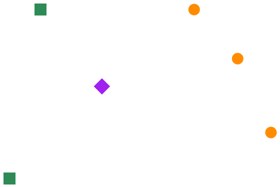
- $\mathcal{P}$: A set of points in $\mathbb{R}^2$



- **Farthest** (point) **Voronoi diagram**
- **Farthest Voronoi region** of site s: $\{x \in \mathbb{R}^2 \mid d(x,s) > d(x,t)\ \forall t \in \mathcal{P}\}$

Color Voronoi Diagrams

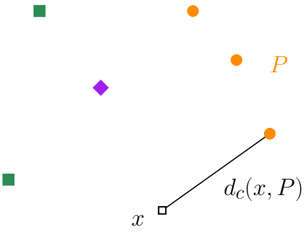
- $\mathcal{P}$: A set of m **clusters of points**, with n overall points.
  $\rightarrow$ Each cluster has a color.



Color Voronoi Diagrams

- $\mathcal{P}$: A set of m **clusters of points**, with n overall points.
  $\rightarrow$ Each cluster has a color.

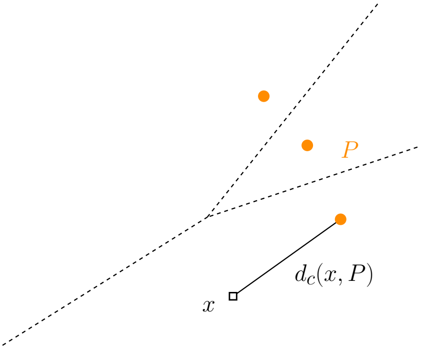
- **Distance** from point x to cluster P: $d_c(x, P) = \min_{p \in P}(x, p)$.



Color Voronoi Diagrams

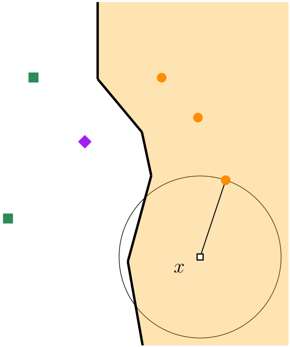
- $\mathcal{P}$: A set of m **clusters of points**, with n overall points.
  $\rightarrow$ Each cluster has a color.

- **Distance** from point x to cluster P: $d_c(x, P) = \min_{p \in P}(x, p)$.



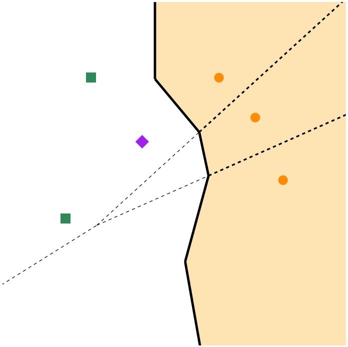
Nearest Color Voronoi Diagram (NCVD)

- The **nearest color region** of a cluster $P \in \mathcal{P}$ is:
  $\{x \in \mathbb{R}^2 \mid d_c(x, P) < d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$



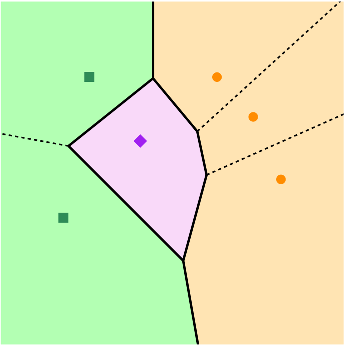
Nearest Color Voronoi Diagram (NCVD)

- The **nearest color region** of a cluster $P \in \mathcal{P}$ is:
$\{x \in \mathbb{R}^2 \mid d_c(x, P) < d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$



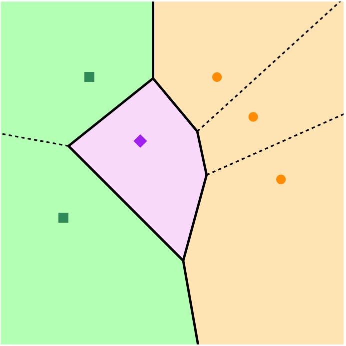
Nearest Color Voronoi Diagram (NCVD)

- The **nearest color region** of a cluster $P \in \mathcal{P}$ is: $\{x \in \mathbb{R}^2 \mid d_c(x, P) < d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$
- **Min-Min** diagram (nearest cluster, nearest distance).



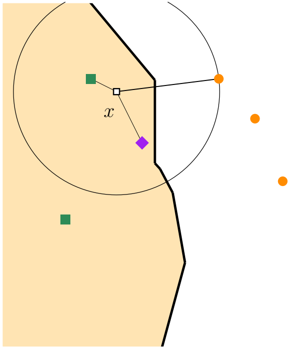
Nearest Color Voronoi Diagram (NCVD)

- The **nearest color region** of a cluster $P \in \mathcal{P}$ is: $\{x \in \mathbb{R}^2 \mid d_c(x, P) < d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$
- **Min-Min** diagram (nearest cluster, nearest distance).
- Nearest point VD $\Rightarrow$ $O(n)$ complexity, $O(n \log n)$ algorithms.



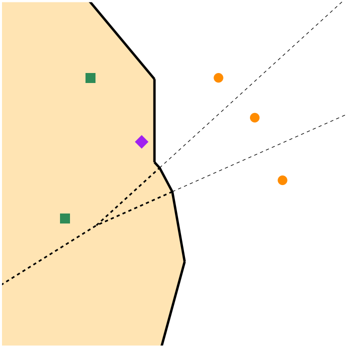
Farthest Color Voronoi Diagram (FCVD)

- The **farthest color region** of a cluster $P \in \mathcal{P}$ is:
  $\{x \in \mathbb{R}^2 \mid d_c(x, P) > d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$



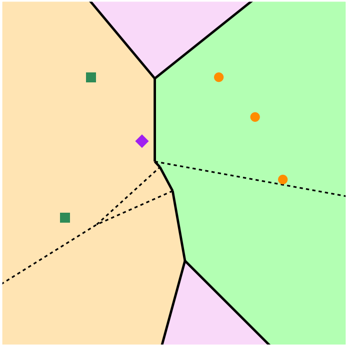
Farthest Color Voronoi Diagram (FCVD)

- The **farthest color region** of a cluster $P \in \mathcal{P}$ is:
  $\{x \in \mathbb{R}^2 \mid d_c(x, P) > d_c(x, Q), \ \forall Q \in \mathcal{P} \setminus \{P\}\}$



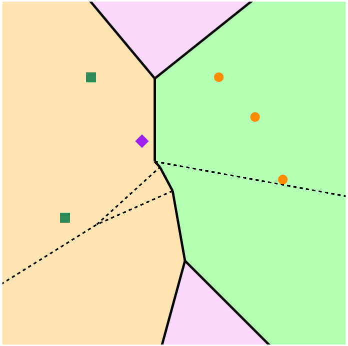
Farthest Color Voronoi Diagram (FCVD)

- The **farthest color region** of a cluster $P \in \mathcal{P}$ is:
  $\{x \in \mathbb{R}^2 \mid d_c(x, P) > d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$



Farthest Color Voronoi Diagram (FCVD)

- The **farthest color region** of a cluster $P \in \mathcal{P}$ is:
  $\{x \in \mathbb{R}^2 \mid d_c(x, P) > d_c(x, Q),\ \forall Q \in \mathcal{P} \setminus \{P\}\}$
- **Max-Min** diagram (farthest cluster, nearest distance).



FCVD - Known results

- Combinatorial **complexity**:
  Upper bound $O(mn)$ [Abellanas et al. 2006]
  Worst case lower bound $\Omega(mn)$ [Huttenlocher et al. 1993]

m : number of clusters n : total number of points

FCVD - Known results

- Combinatorial **complexity**:

  Upper bound $O(mn)$ [Abellanas et al. 2006]

  Worst case lower bound $\Omega(mn)$ [Huttenlocher et al. 1993]
- Construction **algorithms**:

  $O(mn \log n)$ time [Huttenlocher et al. 1993]

m : number of clusters n : total number of points

FCVD - Known results

- Combinatorial **complexity**:
  Upper bound $O(mn)$ [Abellanas et al. 2006]
  Worst case lower bound $\Omega(mn)$ [Huttenlocher et al. 1993]
- Construction **algorithms**:
  $O(mn \log n)$ time [Huttenlocher et al. 1993]
- Special cases:
  [Bae 2012, Claverol et al. 2017, Iacono et al. 2017]

m : number of clusters n : total number of points

FCVD - Known results

- Combinatorial **complexity**:
  Upper bound $O(mn)$ [Abellanas et al. 2006]
  Worst case lower bound $\Omega(mn)$ [Huttenlocher et al. 1993]
- Construction **algorithms**:
  $O(mn \log n)$ time [Huttenlocher et al. 1993]
  $O(n^2)$ time [Edelsbrunner et al. 1989]
- Special cases:
  [Bae 2012, Claverol et al. 2017, Iacono et al. 2017]

m : number of clusters n : total number of points

Motivation - Applications

- **Facility location** with multiple types of facilities.
  Minimum color spanning circle. [Abellanas et al. 2006]
- **Minimum Hausdorff distance** between two sets of points. [Huttenlocher et al. 1993]
- Euclidean Bottleneck **Steiner tree**. [Bae et al. 2010]
- **Sensor deployment** in wireless networks. [Lee et al. 2010]
- **Stabbing circles** for segments. [Claverol et al. 2017]

Hausdorff Voronoi Diagram

- **Min-Max** diagram (nearest cluster, farthest distance).
  The *"dual"* of the FCVD.

Hausdorff Voronoi Diagram

- **Min-Max** diagram (nearest cluster, farthest distance).

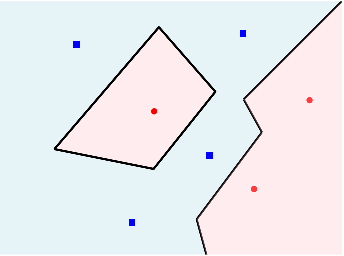
  The *"dual"* of the FCVD.

  Extensively studied:

  → Envelopes in 3 dimensions [Edelsbrunner et al. 1989]
  → Divide and Conquer [Papadopoulou & Lee 2004]
  → Plane Sweep [Papadopoulou 2004]
  → Randomized Incremental [Arseneva & Papadopoulou 2019]

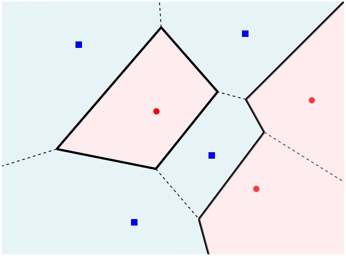
Color bisectors

- The **color bisector** of clusters P and Q is: $b_c(P,Q) = \{x \in \mathbb{R}^2 \mid d_c(x,P) = d_c(x,Q)\}$



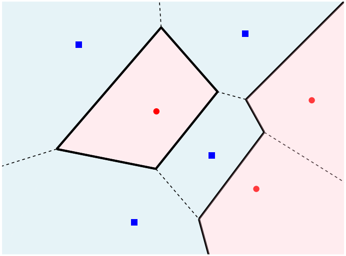
Color bisectors

- The **color bisector** of clusters P and Q is:
  $b_c(P, Q) = \{x \in \mathbb{R}^2 \mid d_c(x, P) = d_c(x, Q)\}$
- $b_c(P, Q)$ is a **subgraph of the diagram** $VD(P \cup Q)$.



Color bisectors

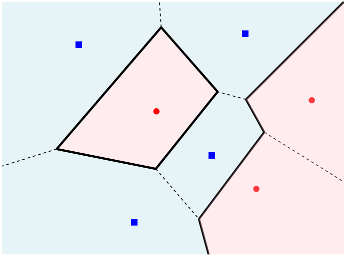
- The **color bisector** of clusters P and Q is:
  $b_c(P, Q) = \{x \in \mathbb{R}^2 \mid d_c(x, P) = d_c(x, Q)\}$
- $b_c(P, Q)$ is a **subgraph of the diagram** $VD(P \cup Q)$.



$\rightarrow$ It consists of bounded and unbounded components.

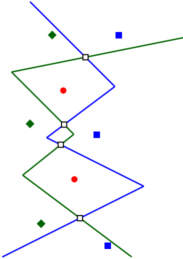
Color bisectors

- The **color bisector** of clusters P and Q is: $b_c(P,Q) = \{x \in \mathbb{R}^2 \mid d_c(x,P) = d_c(x,Q)\}$
- $b_c(P,Q)$ is a **subgraph of the diagram** $VD(P \cup Q)$.



→ It consists of bounded and unbounded components.

→ Two bisectors may intersect linearly many times.

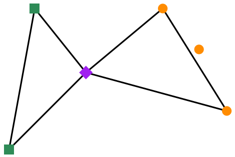


Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

- $O(n)$ size - $O(n \log n)$ time construction.

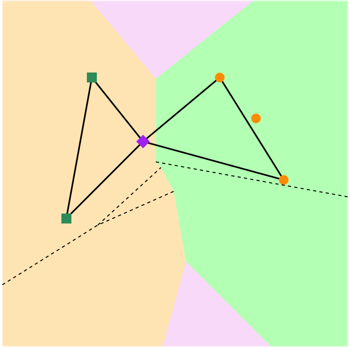
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

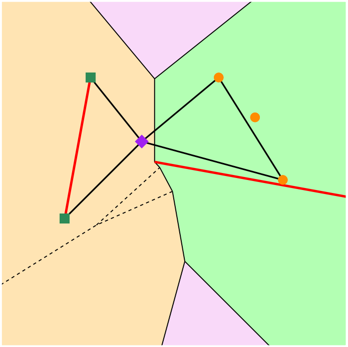
- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

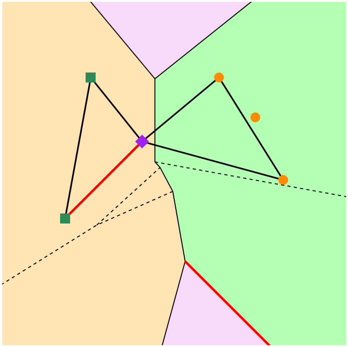
- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

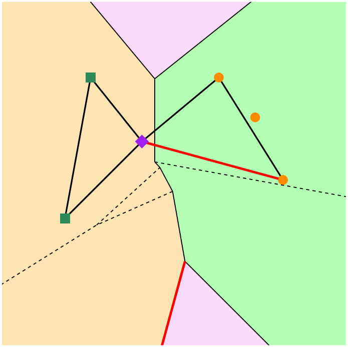
- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

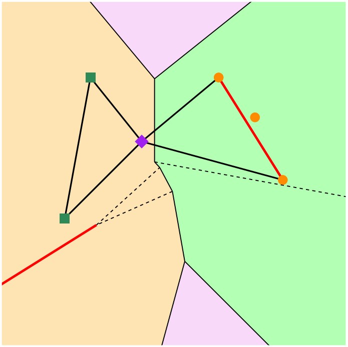
- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

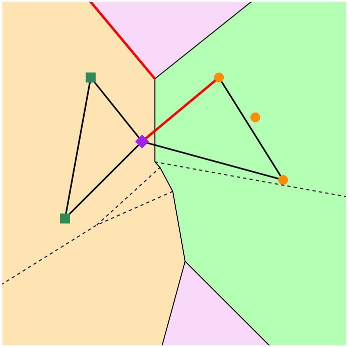
- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

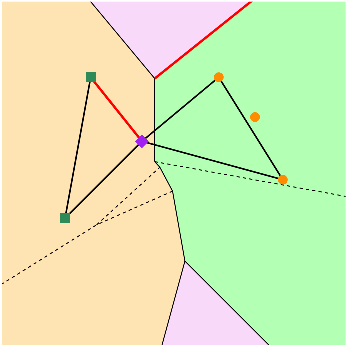
- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

- $O(n)$ size - $O(n \log n)$ time construction.
  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).

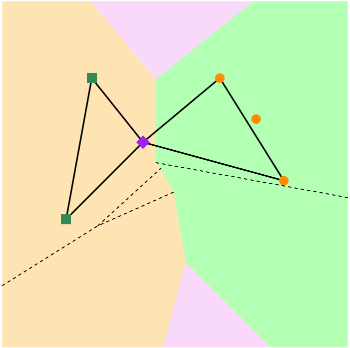


Cluster Hull

The **Cluster Hull** is a closed (non-simple) **polygonal chain**.

- $O(n)$ size - $O(n \log n)$ time construction.

  Defined for the Hausdorff VD [Papadopoulou & Lee 2004]
- Characterizes also the **unbounded faces** of FCVD($\mathcal{P}$).



Combinatorial complexity

What is the **complexity of FCVD($\mathcal{P}$)?**
Need to determine the number of **bounded** and **unbounded faces**.

Combinatorial complexity

What is the **complexity of FCVD($\mathcal{P}$)?**
Need to determine the number of **bounded** and **unbounded faces**.

- **Unbounded** faces?

Combinatorial complexity

What is the **complexity of FCVD($\mathcal{P}$)?**
Need to determine the number of **bounded** and **unbounded faces**.

- **Unbounded** faces?

  FCVD($\mathcal{P}$) has $O(n)$ unbounded faces.

  Key: Cluster hulls.

Combinatorial complexity

What is the **complexity of FCVD($\mathcal{P}$)?**
Need to determine the number of **bounded** and **unbounded faces**.

- **Unbounded** faces?

  FCVD($\mathcal{P}$) has $O(n)$ unbounded faces.

  Key: Cluster hulls.

- **Bounded** faces?

Combinatorial complexity

What is the **complexity of FCVD($\mathcal{P}$)?**
Need to determine the number of **bounded** and **unbounded faces**.

- **Unbounded** faces?

  FCVD($\mathcal{P}$) has $O(n)$ unbounded faces.

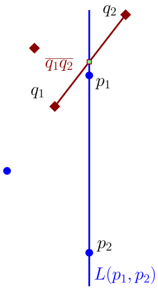
  Key: Cluster hulls.

- **Bounded** faces?

  Key: Define **straddles**.

Straddles

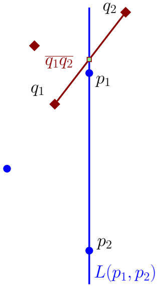
Cluster Q (or $q_1, q_2 \in Q$), **straddles** $p_1, p_2 \in P$ if:
$\rightarrow \overline{q_1 q_2}$ intersects $L(p_1, p_2)$.



Straddles

Cluster Q (or $q_1, q_2 \in Q$), **straddles** $p_1, p_2 \in P$ if:
$\rightarrow$ $\overline{q_1 q_2}$ intersects $L(p_1, p_2)$.

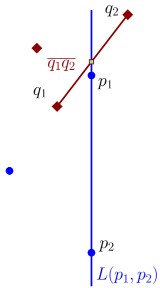
$s(p_1, p_2)$: number of clusters that straddle p_1, p_2.



Straddles

Cluster Q (or $q_1, q_2 \in Q$), **straddles** $p_1, p_2 \in P$ if:
$\rightarrow$ $\overline{q_1 q_2}$ intersects $L(p_1, p_2)$.

$s(p_1, p_2)$: number of clusters that straddle p_1, p_2.

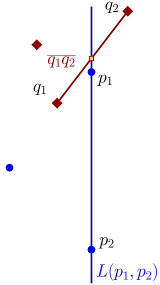


- $s(\mathcal{P})$: **straddling number** of $\mathcal{P}$
- $s(\mathcal{P}) = \sum_{P_i \in \mathcal{P}} \sum_{(p_j, p_k) \in P_i} s(p_j, p_k)$

Straddles

Cluster Q (or $q_1, q_2 \in Q$), **straddles** $p_1, p_2 \in P$ if:
$\rightarrow \overline{q_1 q_2}$ intersects $L(p_1, p_2)$.

$s(p_1, p_2)$: number of clusters that straddle p_1, p_2.

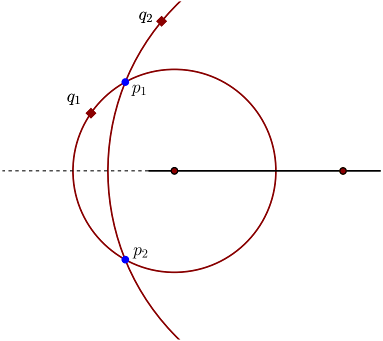


- $s(\mathcal{P})$: **straddling number** of $\mathcal{P}$
- $s(\mathcal{P}) = \sum_{P_i \in \mathcal{P}} \sum_{(p_j, p_k) \in P_i} s(p_j, p_k)$
- $s(\mathcal{P}) = O(mn)$

Straddles

Cluster Q (or $q_1, q_2 \in Q$), **straddles** $p_1, p_2 \in P$ if:
$\rightarrow \overline{q_1 q_2}$ intersects $L(p_1, p_2)$.

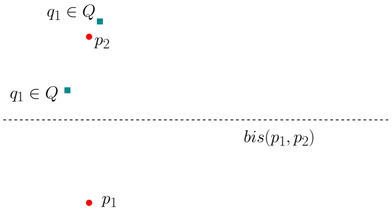
$s(p_1, p_2)$: number of clusters that straddle p_1, p_2.



- $s(\mathcal{P})$: **straddling number** of $\mathcal{P}$
- $s(\mathcal{P}) = \sum_{P_i \in \mathcal{P}} \sum_{(p_j, p_k) \in P_i} s(p_j, p_k)$
- $s(\mathcal{P}) = O(mn)$

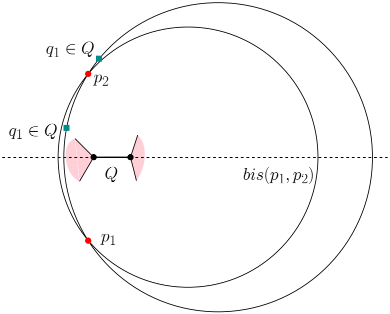
Refined combinatorial complexity

- FCVD($\mathcal{P}$) has $O(n + s(\mathcal{P}))$ bounded faces.



Refined combinatorial complexity

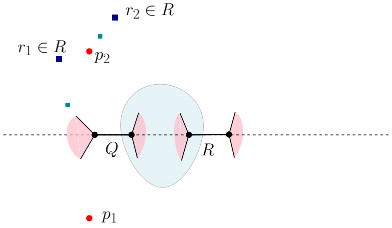
- FCVD($\mathcal{P}$) has $O(n + s(\mathcal{P}))$ bounded faces.



Refined combinatorial complexity

- FCVD($\mathcal{P}$) has $O(n + s(\mathcal{P}))$ bounded faces.

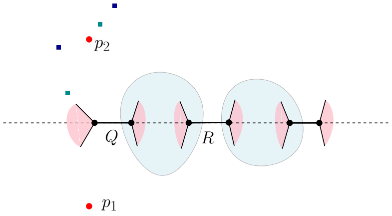
  Key: They **appear** on edges of the internal subdivision **consecutively** induced by distinct clusters.



Refined combinatorial complexity

- FCVD($\mathcal{P}$) has $O(n + s(\mathcal{P}))$ bounded faces.

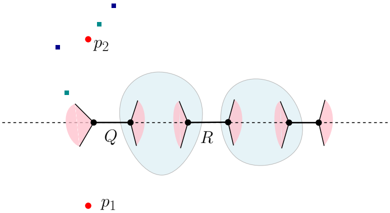
  Key: They **appear** on edges of the internal subdivision **consecutively** induced by distinct clusters.



Refined combinatorial complexity

- FCVD($\mathcal{P}$) has $O(n + s(\mathcal{P}))$ bounded faces.

  Key: They **appear** on edges of the internal subdivision **consecutively** induced by distinct clusters.



Theorem - Combinatorial complexity

FCVD($\mathcal{P}$) has $O(n + s(\mathcal{P}))$ complexity.

Note: Refine $O(mn)$ upper bound [Abellanas et al. 2006].

Conditions for linear-size diagrams

If FCVD($\mathcal{P}$) has $\Theta(n^2)$ size, the $O(n^2)$ algorithm is optimal.

We are interested in:

- **Conditions for $O(n)$ combinatorial complexity.**
- $o(n^2)$-**time algorithms** for these cases.

Admissible cluster 1/2

$\mathcal{P}$ is **admissible**, if it falls under the framework of **Abstract** Voronoi diagrams [Klein 1989].

Admissible cluster 1/2

$\mathcal{P}$ is **admissible**, if it falls under the framework of **Abstract** Voronoi diagrams [Klein 1989].

$\mathcal{P}$ is admissible, if for every $\mathcal{P}' \subseteq \mathcal{P}$:

1) Color **bisectors** are **unbounded**.
2) **Nearest** color **regions** are non-empty and **connected**.
3) The union of all nearest color **regions** covers the plane.

Admissible cluster 1/2

$\mathcal{P}$ is **admissible**, if it falls under the framework of **Abstract** Voronoi diagrams [Klein 1989].

$\mathcal{P}$ is admissible, if for every $\mathcal{P}' \subseteq \mathcal{P}$:

1) Color **bisectors** are **unbounded**.
2) **Nearest** color **regions** are non-empty and **connected**.
3) The union of all nearest color **regions** covers the plane.

- If $\mathcal{P}$ is admissible, FCVD($\mathcal{P}$) is a tree of $O(n)$ complexity. Follows [Mehlhorn et al. 2001]

Admissible cluster 2/2

Theorem - Necessary & sufficient condition

$\mathcal{P}$ is admissible if and only if:
(1) each region in NCVD($\mathcal{P}$) is connected.
(2) no cluster is contained in the convex hull of another cluster.

Key: Check region connectivity **only for** $\mathcal{P}$.

Admissible cluster 2/2

Theorem - Necessary & sufficient condition

$\mathcal{P}$ is admissible if and only if:
(1) each region in NCVD($\mathcal{P}$) is connected.
(2) no cluster is contained in the convex hull of another cluster.

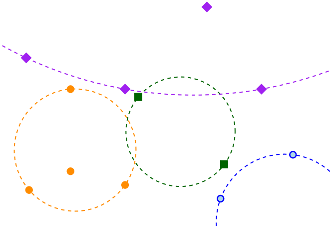
Key: Check region connectivity **only for** $\mathcal{P}$.

- If $\mathcal{P}$ is linearly separable, we can decide if $\mathcal{P}$ is admissible in $O(n \log n)$ time.

  Key: Check region connectivity using NCVD($\mathcal{P}$).

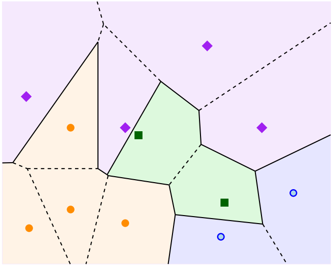
Disk-separable clusters

$\mathcal{P}$ is **disk-separable** if:
for any $P \in \mathcal{P}$ there exists a disk containing P and no other point.



Disk-separable clusters

$\mathcal{P}$ is **disk-separable** if:
for any $P \in \mathcal{P}$ there exists a disk containing P and no other point.



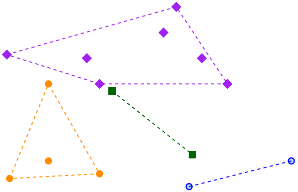
Theorem - Sufficient condition

If $\mathcal{P}$ is disk-separable, then $\mathcal{P}$ is admissible.

Key: Disk-separability implies region connectivity.

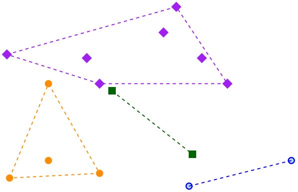
Linearly separable clusters

$\mathcal{P}$ is **linearly separable** if:
clusters have pairwise disjoint convex hulls.



Linearly separable clusters

$\mathcal{P}$ is **linearly separable** if:
clusters have pairwise disjoint convex hulls.

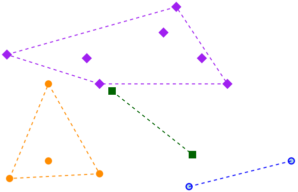


Is linear separability a condition for $\mathcal{P}$:

- to be admissible?

Linearly separable clusters

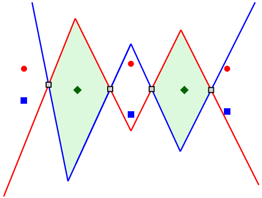
$\mathcal{P}$ is **linearly separable** if:
clusters have pairwise disjoint convex hulls.



Is linear separability a condition for $\mathcal{P}$:

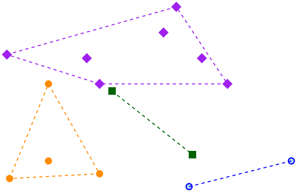
- to be admissible?

  No.



Linearly separable clusters

$\mathcal{P}$ is **linearly separable** if:
clusters have pairwise disjoint convex hulls.

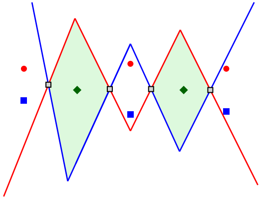


Is linear separability a condition for $\mathcal{P}$:

- to be admissible?

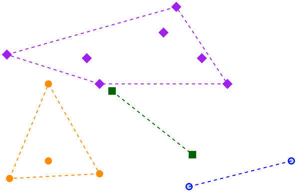
  No.

- to have $O(n)$ size?



Linearly separable clusters

$\mathcal{P}$ is **linearly separable** if:
clusters have pairwise disjoint convex hulls.



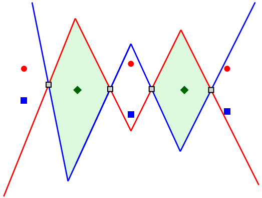
Is linear separability a condition for $\mathcal{P}$:

- to be admissible?

  No.

- to have $O(n)$ size?

  No.



Lower bound Construction 1/2

We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Lower bound Construction 1/2

We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Intuitional description:

- Cluster $L_i, U_i \to$ Segment $\overline{L_i U_i}$

Lower bound Construction 1/2

We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Intuitional description:

- Cluster $L_i, U_i \rightarrow$ Segment $\overline{L_i U_i}$
- $\overline{L_1 U_1}$: vertical segment of length 2^m

Lower bound Construction 1/2

We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Intuitional description:

- Cluster $L_i, U_i \rightarrow$ Segment $\overline{L_i U_i}$
- $\overline{L_1 U_1}$: vertical segment of length 2^m
- $\overline{L_i U_i}$: vertical segment with $U_i = U_{i-1}$
- Halve length and translate upwards.

Lower bound Construction 1/2

We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Intuitional description:

- Cluster $L_i, U_i \rightarrow$ Segment $\overline{L_i U_i}$
- $\overline{L_1 U_1}$: vertical segment of length 2^m
- $\overline{L_i U_i}$: vertical segment with $U_i = U_{i-1}$
- Halve length and translate upwards.

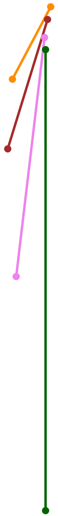
Lower bound Construction 1/2

We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Intuitional description:

- Cluster $L_i, U_i \to$ Segment $\overline{L_i U_i}$
- $\overline{L_1 U_1}$: vertical segment of length 2^m
- $\overline{L_i U_i}$: vertical segment with $U_i = U_{i-1}$
- Halve length and translate upwards.

Lower bound Construction 1/2



We construct a family $\mathcal{P} = \{P_i = \{l_i, u_i\}\}$.
L_i: *lower* point U_i: *upper* point.

Intuitional description:

- Cluster $L_i, U_i \rightarrow$ Segment $\overline{L_i U_i}$
- $\overline{L_1 U_1}$: vertical segment of length 2^m
- $\overline{L_i U_i}$: vertical segment with $U_i = U_{i-1}$
- Halve length and translate upwards.
- Perform some *"rotation"* around upper points

Lower bound Construction 2/2

Properties of the constructed set $\mathcal{P}$

- $\mathcal{P}$ is linearly separable
- $s(\mathcal{P}) = \Theta(m^2)$
- Every straddle induces a vertex to FCVD($\mathcal{P}$).

Lower bound Construction 2/2

Properties of the constructed set $\mathcal{P}$

- $\mathcal{P}$ is linearly separable
- $s(\mathcal{P}) = \Theta(m^2)$
- Every straddle induces a vertex to FCVD($\mathcal{P}$).

Combining $\Omega(m^2)$ with trivial $\Omega(n)$ bound:

Theorem - Lower bound

If $\mathcal{P}$ is linearly separable, FCVD($\mathcal{P}$) has $\Omega(n + m^2)$ complexity in the worst case.

Algorithm description

Divide & Conquer algorithm:

1. **Divide** $\mathcal{P}$ in two sets $\mathcal{P}_A$ and $\mathcal{P}_B$.
2. **Recursively compute** FCVD($\mathcal{P}_A$) and FCVD($\mathcal{P}_B$).
3. **Merge** FCVD($\mathcal{P}_A$) and FCVD($\mathcal{P}_B$) into FCVD($\mathcal{P}_A \cup \mathcal{P}_B$).

Algorithm description

Divide & Conquer algorithm:

1. **Divide** $\mathcal{P}$ in two sets $\mathcal{P}_A$ and $\mathcal{P}_B$.
2. **Recursively compute** FCVD($\mathcal{P}_A$) and FCVD($\mathcal{P}_B$).
3. **Merge** FCVD($\mathcal{P}_A$) and FCVD($\mathcal{P}_B$) into FCVD($\mathcal{P}_A \cup \mathcal{P}_B$).

Construct the **merge curve**.

$\rightarrow$ Can have many components.

$\rightarrow$ Can be bounded and unbounded.

Constructing the merge curve

For each component:

a. Find a **starting point**.

b. **Trace** the component.

Constructing the merge curve

For each component:

a. Find a **starting point**.

b. **Trace** the component.

- **Tracing** a component takes linear time.

  Key: Using a *visibility property*

Constructing the merge curve

For each unbounded component:

a. **Find a starting point.**

b. **Trace** the component.

- Finding **starting points** on **unbounded** components takes $O(n)$ time at each step.

  Key: Merging cluster hulls before merging diagrams, similar to [Papadopoulou & Lee 2004].

Starting points on bounded components

For each bounded component:

a. Find a **starting point**.

Key: The internal subdivision of every bounded face is a tree.
Need to search for edges of the internal subdivision.

Starting points on bounded components

For each bounded component:

a. Find a **starting point**.

Key: The internal subdivision of every bounded face is a tree.
Need to search for edges of the internal subdivision.

- $\rightarrow$ Use data structure of [Iacono et al. 2017]
- $\rightarrow$ $O(n \log n)$ time to build at each step.
- $\rightarrow$ For each occurrence of an edge: $O(\log^2 n)$ search procedure.

Algorithm: General case

Theorem - General algorithm

FCVD($\mathcal{P}$) can be constructed in $O((n + s(\mathcal{P})) \log^3 n)$ time.

Key: Make $O(\log^2 n)$ search only for potential bounded faces.
Note: Faster than existing algorithms, if $s(\mathcal{P}) = O(n)$.

Algorithm: Admissible clusters

Theorem - Admissible clusters

If $\mathcal{P}$ is admissible, FCVD($\mathcal{P}$) can be constructed in $O(n \log n)$ time.

Key: FCVD($\mathcal{P}$) is a tree, so no bounded components.

Conclusions

Farthest Color Voronoi Diagrams

Conclusions

Farthest Color Voronoi Diagrams

- Refined **combinatorial complexity**: **straddles**.

Conclusions

Farthest Color Voronoi Diagrams

- Refined **combinatorial complexity**: **straddles**.
- **Conditions** under which FCVD has **linear complexity**.

Conclusions

Farthest Color Voronoi Diagrams

- Refined **combinatorial complexity**: **straddles**.
- **Conditions** under which FCVD has **linear complexity**.
- **Linear separability**: quadratic **lower bound**.

Conclusions

Farthest Color Voronoi Diagrams

- Refined **combinatorial complexity**: **straddles**.
- **Conditions** under which FCVD has **linear complexity**.
- **Linear separability**: quadratic **lower bound**.
- **Construction algorithms**: $O((n + s(\mathcal{P})) \log^3 n)$.

Thank you for your attention!

Questions?