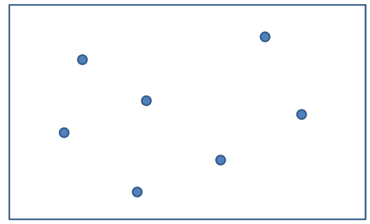
The Spatial Skyline Queries

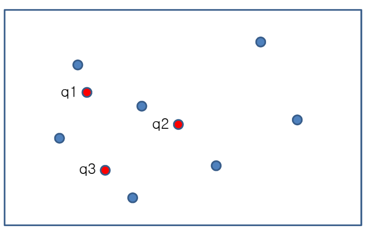
Mehdi Sharifzadeh and Cyrus Shahabi
VLDB 2006

Presented by
Ali Khodaei

- Coffee shop

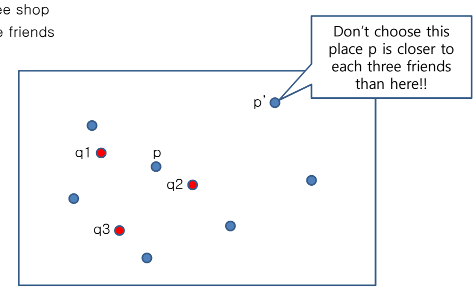


- Coffee shop
- Three friends



- query point

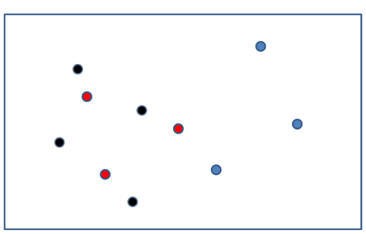
- Coffee shop
- Three friends



- query point

dist(p,q1) < dist(p',q1) and
dist(p,q2) < dist(p',q2) and
dist(p,q3) < dist(p',q3)

➔ p spatially dominate p'



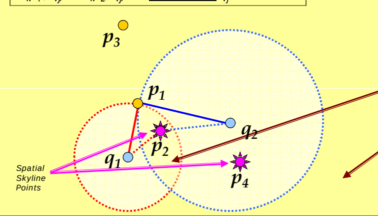
- query point
- spatial skyline point

spatial skyline point is not dominated by any other data point

---

Problem Definition

p_1 **spatially dominates** p_2 *with respect to Q iff*
$D(p_1, q_i) \leq D(p_2, q_i)$ for <u>all</u> q_i in Q and
$D(p_1, q_j) < D(p_2, q_j)$ for <u>at least one</u> q_j



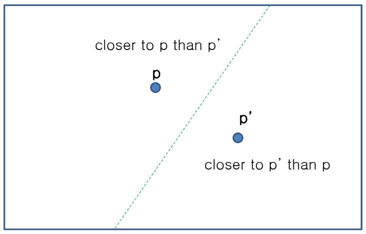
- *Data P = {p_1, p_2, p_3, p_4}*
- *Query Q = {q_1, q_2}*
- *Distance D() = Euclidean*
- *p_2 **spatially dominates** p_1 with respect to {q_1, q_2}*
- ***Dominator Region of** p_1*
- *p_1 **spatially dominates** p_3*
- ***Dominance Region** of p_1*
- *No dominance relation between p_1 and p_4*

Spatial Skyline Query (SSQ): find the data points p_i that are **<u>not</u>** spatially dominated by any other point p_j with respect to the given query points (here, p_2 and p_4).

□ spatially domiante

p spatially dominate p′ if p is closer to All query points than p′

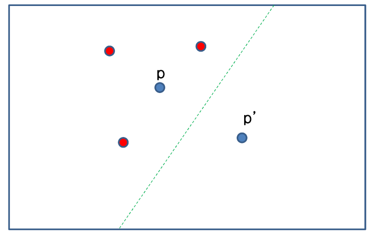
if $D(p, q_i) \le D(p', q_i)$ for all $q_i \in Q$



□ spatially domiante

p spatially dominate p′

if $D(p, q_i) \le D(p', q_i)$ for all $q_i \in Q$

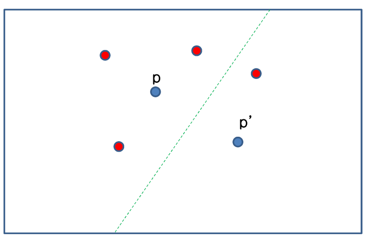


p spatially dominate p'

□ spatially domiante

p spatially dominate p′

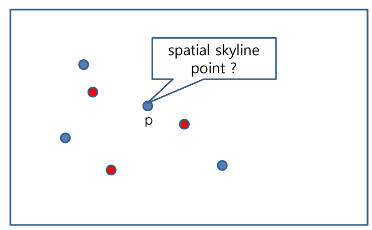
if $D(p, q_i) \le D(p', q_i)$ for all $q_i \in Q$



p can't spatially dominate p'

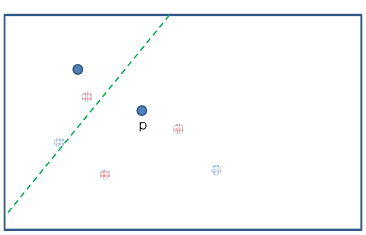
□ Spatial skyline

Spatial skyline point : p is not spatially dominated by any other point in P than p is spatial skyline point



□ Spatial skyline

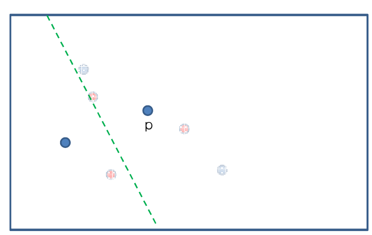
Spatial skyline point : not spatially dominated by any other point in P



O(|Q|)

□ Spatial skyline

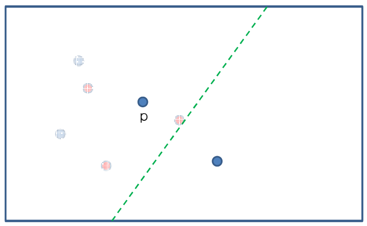
Spatial skyline point : not spatially dominated by any other point in P



O(|Q|)

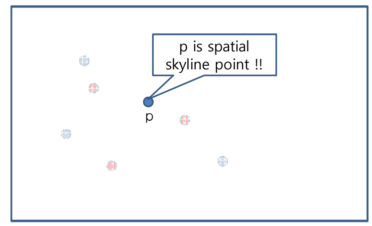
☐ Spatial skyline

Spatial skyline point : not spatially dominated by any other point in P

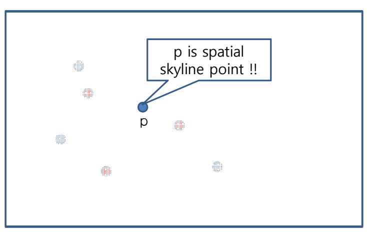


☐ Spatial skyline

Spatial skyline point : not spatially dominated by any other point in P



☐ Spatial skyline

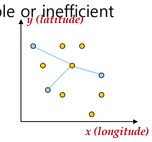


we can compute all spatial skyline points

Time Complexity: O(|P|² |Q|)
|P|: number of data points, |Q|: number of query points

(naïve algorithm)

- **Naïve approach**
  - **Complexity:** *O(|P|² |Q|)*
    |P|: number of data points, *|Q|*: number of query points
- **Why a new algorithm is needed:**
  - Complexity of Naïve approach is high
    - Each dominance check involves *2|Q|* distance computation operations: increases with more query points
  - General skyline algorithms are either inapplicable or inefficient
    - Due to dynamic spatial attributes
  - Optimization opportunity
    - The geometric properties of space can be exploited

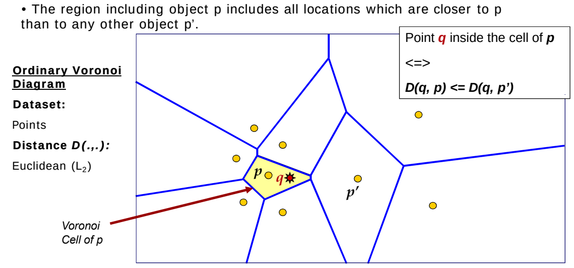


Geometric Properties

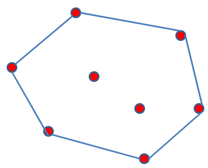
- **Complexity of Naïve approach:** *O(|P|² |Q|)*
  - *|P|*: number of data points
  - *|Q|*: number of query points
- We identify geometric properties to reduce th is complexity by reducing the number of :
  - data points to be investigated
  - query points that has no effect on the result
- Less and cheaper dominance checks
- We identify three properties …

Preliminaries:Voronoi Diagras

- Given a set of spatial objects, a Voronoi diagram *uniquely* partitions the space into disjoint regions (cells).
- The region including object p includes all locations which are closer to p than to any other object p'.



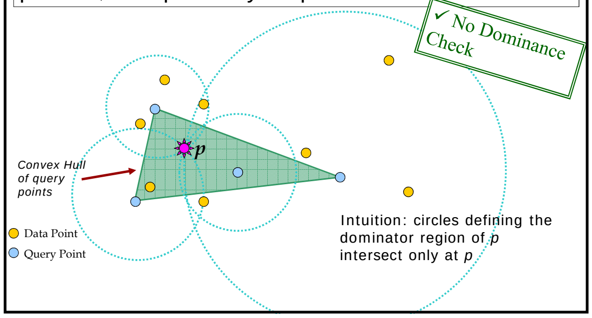
Preliminaries: Convex Hull



Convex hull of a set Q of points is the smallest convex polygon CH for which each point in Q is either on the boundary of CH or in its interior

Geometric Properties

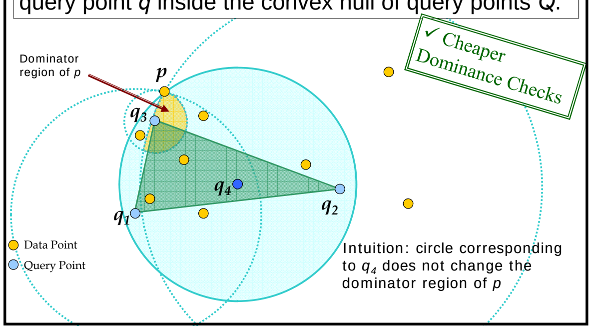
GP$_1$: Any point *p* inside the **convex hull** of query points *Q* is a spatial skyline point.



Intuition: circles defining the dominator region of *p* intersect only at *p*

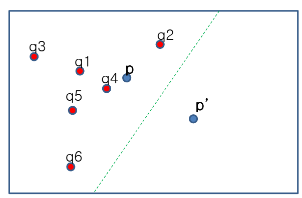
Geometric Properties

GP$_2$: The set of skyline points does not depend on any query point *q* inside the convex hull of query points *Q*.

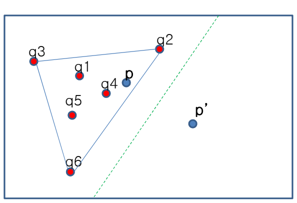


Intuition: circle corresponding to q_4 does not change the dominator region of *p*

naive algorithm



compute whether q1 is in left side or right side
compute whether q2 is in left side or right side
compute whether q3 is in left side or right side
compute whether q4 is in left side or right side
compute whether q5 is in left side or right side



compute q2 is in left side or right side
compute q3 is in left side or right side
compute q6 is in left side or right side

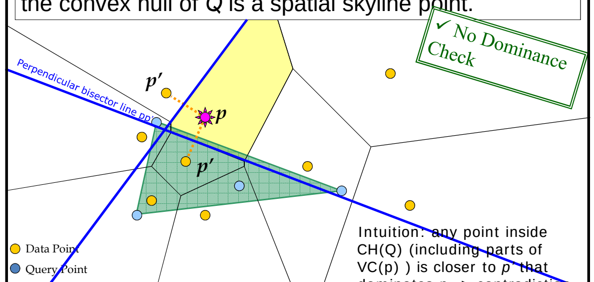
compute vertices of convex hull are in left side or right side

➔ take O(|CHv(Q)|) time

we don't need to compute q1,q4 and q5

Geometric Properties

GP$_3$: Any point *p* whose Voronoi cell **intersects** with the convex hull of *Q* is a spatial skyline point.



Intuition: any point inside CH(Q) (including parts of VC(p)) is closer to *p'* that dominates *p* -> contradiction

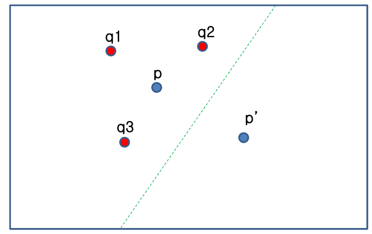
Algorithm: VS²

- **VS²**: Voronoi-based Spatial Skyline Algorithm
- Utilizes the geometric interpretation of the skyline
  - GP1 GP3 – With **no dominance check**, adds any data point *p* whose Voronoi cell intersects with the convex hull of *Q*
  - GP2 – Performs **cheaper dominance check only on a small subset** of points (neighbors of skyline points ~ O(S))
- Traversal is based on monotone function mindist

□ mindist of p : dist(p,q1) + dist(p,q2) + dist(p,q3)

if p dominate p' then mindist(p) < mindist(p')

dist(p,q1) < dist(p',q1) and
dist(p,q2) < dist(p',q2) and
dist(p,q3) < dist(p',q3)
(by definition of 'spatially dominate)



□ mindist of p : dist(p,q1) + dist(p,q2) + dist(p,q3)

if p dominate p' then mindist(p) < mindist(p')

➔ if mindist(p) > mindist(p'), than p can't dominate p'

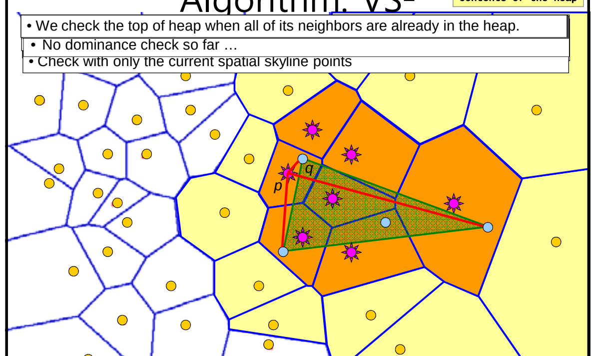
So, sort the data point by mindist ascending order.

To determine whether p is skyline point or not,

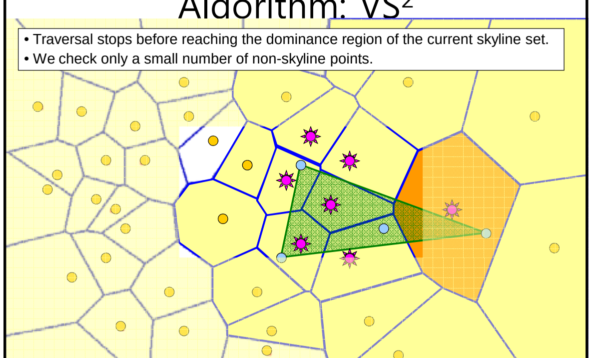
we only need to compare p with the point that has smaller mindist than p.

Algorithm: VS²

Top of the heap
Contents of the heap



Algorithm: VS²



Algorithm: VS²

- **Time Complexity:** $O(|S|^2 |CH_v(Q)| + \Phi(|P|))$
  - Naïve: $O(|P|^2 |Q|)$
- $|S|$: number of skyline points
- $|CH_v(Q)|$: number of vertices of the convex hull of Q ($\le |Q|$)
- $\Phi(|P|)$: complexity of finding the data point from which VS² starts traversing inside the convex hull of Q ($O(\log(|P|))$
- **Space Complexity:** $O(|P|)$
  - *Space required for ordinary Voronoi Diagram is O(|P|)*

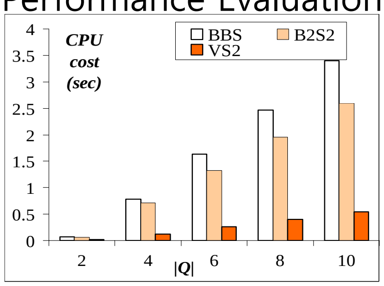
Algorithms: B^2S^2

- **B^2S^2:** Branch-and-Bound Spatial Skyline Algorith m
- Customization of BBS [Papadias et al.] for SSQs
- Uses **some** of the geometric properties of the s kyline (GP_1 and GP_2)
- Similar to BBS traverses an R-tree on data points
- **Traversal order:** specified by any monotone function (e.g., *mindist(p, $CH_v(Q)$)*)

Performance Evaluation

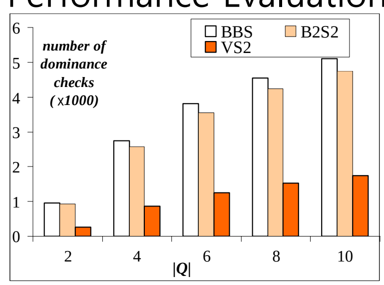
- **Dataset:** USGS including one million loc ations
- R*-tree on data points for BBS and B^2S^2
- Pre-built Delaunay graph of data points for VS^2

Performance Evaluation



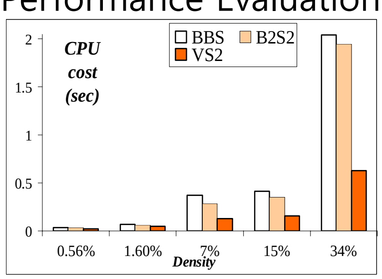
- *Max MBR(Q)=0.3%*
- The difference in improvement of VS^2 over BBS increases for larger query sets.

Performance Evaluation



- Variations of B^2S^2 require less dominance checks than BBS.
- Note that each dominance check is cheaper in our VS^2 and B^2S^2 algorithms.

Performance Evaluation



- *Max |MBR(Q)| = 0.5%, |Q| = 6*
- VS^2 is also scalable with respect to the density of data (i.e., number of skyline points)

Questions?