USC Viterbi
School of Engineering

Time Parameterized Queries in Spatio Temporal Databases

Instructor: Cyrus Shahabi

Outline

- Introduction
- Related work
- TP Queries
  - TP Window Query
  - TP K Nearest Neighbor Query
  - TP Join Query
- Conclusion

Introduction

- Conventional Queries
- Continuous Queries
- Time Parameterized Queries

Conventional Queries

- These are the traditional ‘*instantaneous*’ queries that are evaluated only once to return a single result.
- Are these type of queries reliable in dynamic environments?
  - No!
- Why?

Conventional Queries

- These are the traditional ‘*instantaneous*’ queries that are evaluated only once to return a single result.
- Are these type of queries reliable in dynamic environments?
  - No!
- Why?
  - The results of a conventional query may be invalidated very soon due to the movements of objects and queries
  - E.g., Which are my nearest gas stations now?

Continuous Queries

- What is Continuous Queries?

Continuous Queries

- What is Continuous Queries?
  - Moving objects only
  - Moving queries only
  - Both

Continuous Queries

- What is Continuous Queries?
  - Moving objects only
  - Moving queries only
  - Both
- How to deal with such queries? Updates!

Continuous Queries

- What is Continuous Queries?
  - Moving objects only
  - Moving queries only
  - Both
- How to deal with such queries? Updates!
  - When to update?

Continuous Queries

- What is Continuous Queries?
  - Moving objects only
  - Moving queries only
  - Both
- How to deal with such queries? Updates!
  - When to update?
- Due to their dynamic nature, the result of any query is strongly related to the **temporal** context.

Time Parameterized Queries

Time Parameterized Queries

- Time Parameterized queries (TP) , whenever a query is issued, a TP returns

Time Parameterized Queries

- Time Parameterized queries (TP) , whenever a query is issued, a TP returns
  - The **actual result** that satisfies the corresponding spatial query.

Time Parameterized Queries

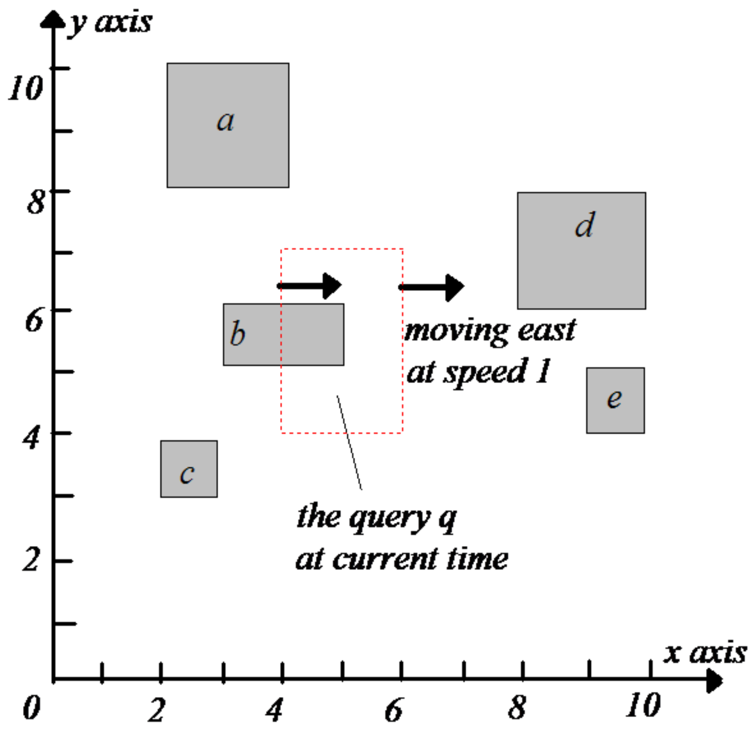
- Time Parameterized queries (TP) , whenever a query is issued, a TP returns
  - The **actual result** that satisfies the corresponding spatial query.
  - The **validity period**/expiration time of the result.

Time Parameterized Queries

- Time Parameterized queries (TP) , whenever a query is issued, a TP returns
  - The **actual result** that satisfies the corresponding spatial query.
  - The **validity period**/expiration time of the result.
  - The **change** that cause the expiration of the results.

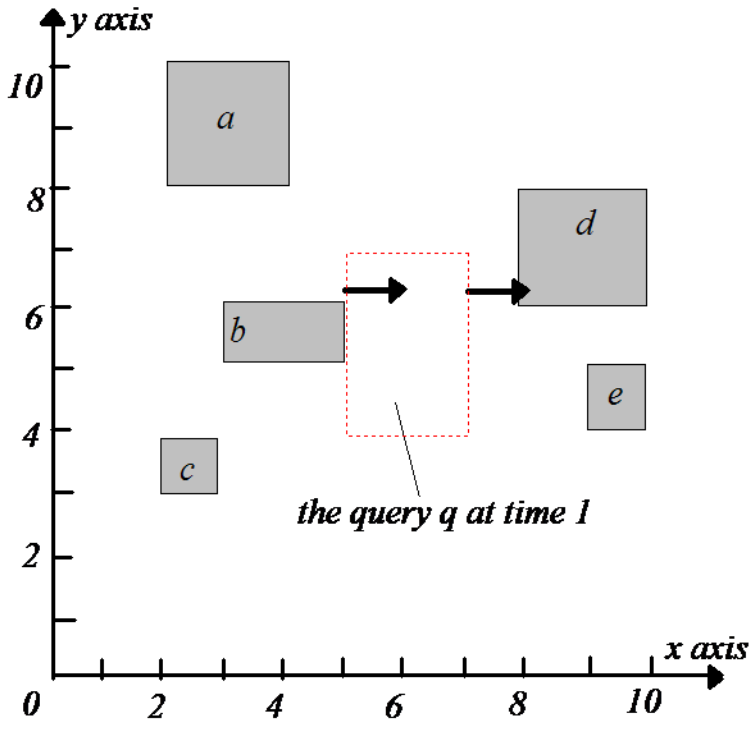
USC Viterbi
School of Engineering

Time Parameterized Queries



Conventional Query

- Result={b}



Time Parameterized Query

At time 1, *b* would be the nearest neighbor. After that time, the results expire, and *d* would be the new nearest neighbor.

Time Parameterized Queries

- Instead of Updates:
  - Here the objects dynamic behavior does not necessarily require updates but can be stored as a function of time using appropriate indexes.
- All TP queries could be reduced to:
  - Some form of NN search
- Could be applied in
  - Moving object
  - Moving query
  - Both

Related Work

- Time Parameterized R-tree (TPR tree)
  - TPR-tree is an extension of an R-tree that can answer prediction queries on dynamic objects.
- Branch and Bound (BaB) Algorithm
  - Using R-tree for NN queries
    - Bound: Min-dist, minmax-dist
  - Different searching manner:
    - Depth first (DF) traversing
    - Best first (BF) traversing

Time Parameterized tree

Key Features:

Time Parameterized tree

Key Features:

- A dynamic object is represented with an MBR that bounds its extents at the current time. Each dynamic object has a velocity vector.

Time Parameterized tree

Key Features:

- A dynamic object is represented with an MBR
  that bounds its extents at the current time.
  Each dynamic object has a velocity vector.

- Future MBRs are not stored explicitly
  but computed based on current extents and velocity vectors

Time Parameterized tree

Key Features:

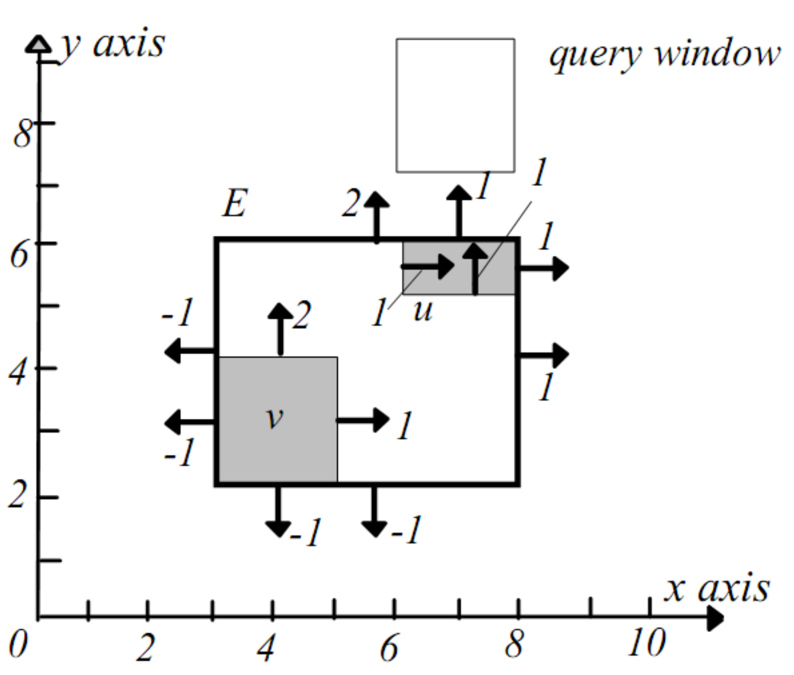
- A dynamic object is represented with an MBR that bounds its extents at the current time. Each dynamic object has a velocity vector.

- Future MBRs are not stored explicitly but computed based on current extents and velocity vectors

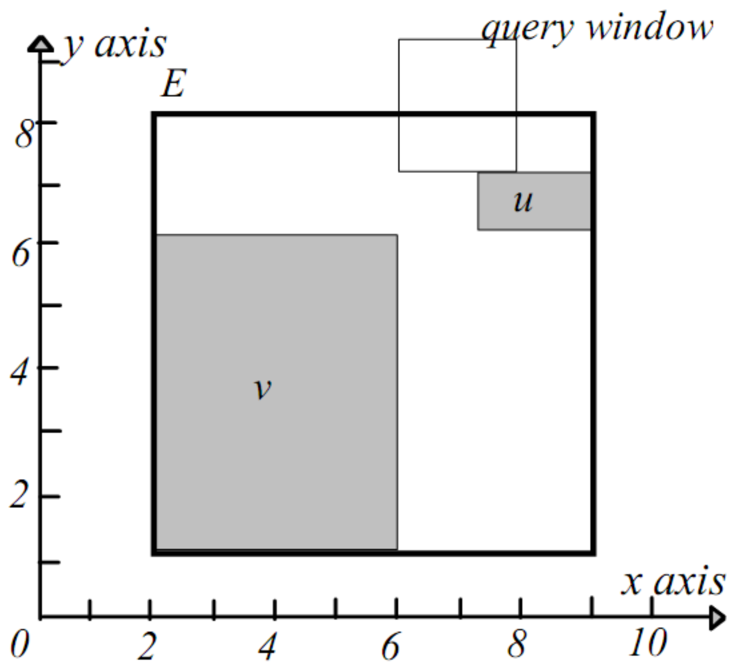
- As in a traditional R-tree, the extents are such that the MBR tightly encloses all entries in the node at current time.

Time Parameterized tree

- Different edge velocities will cause the object to grow or shrink with time.



Time t



Time t+1

Time Parameterized (TP) queries

- Introduction
- TP Window Query
- TP K Nearest Neighbor Query
- TP Join Query

Time Parameterized (TP) queries

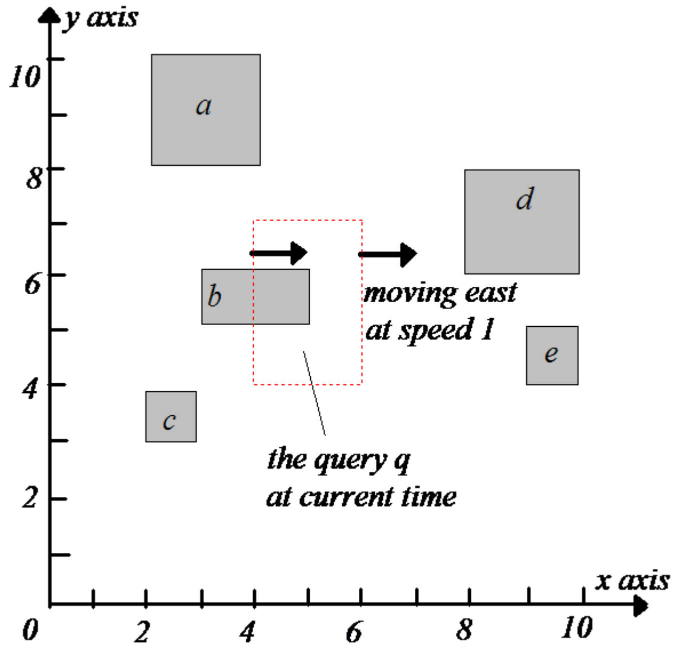
- Introduction
- TP Window Query
- TP K Nearest Neighbor Query
- TP Join Query

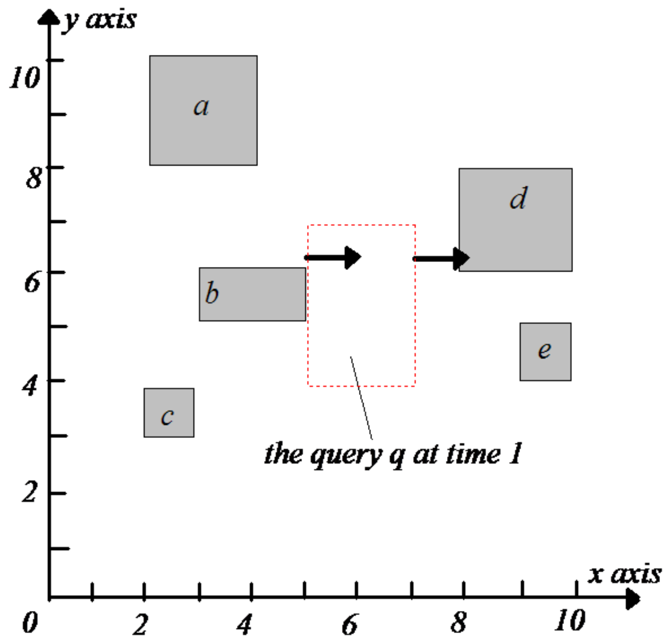
Time Parameterized (TP) queries

- Intro: Output form <R,T,C>
  - R- set of objects satisfying the query (current result)
  - T- is the expiry time of the results
  - C- set of objects that will affect R at time T (<T,C> - TP component)

Time Parameterized (TP) queries

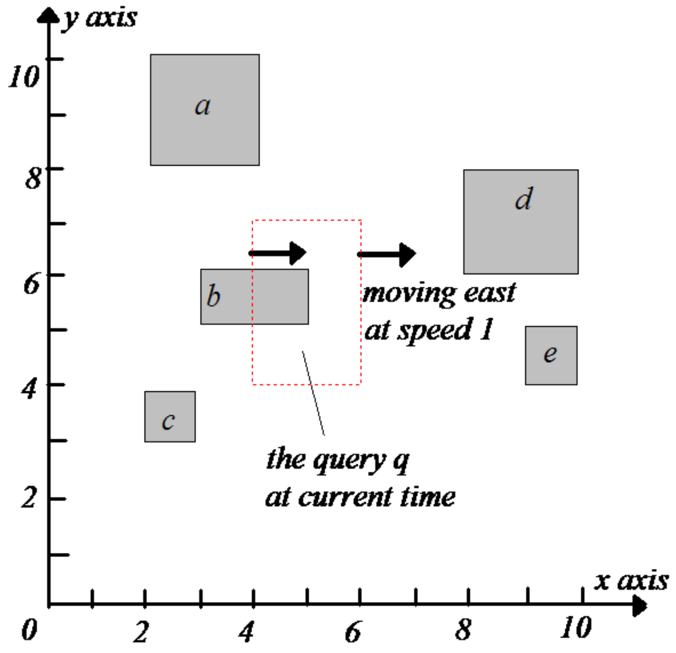
- Intro: Output form <R,T,C>
  - R- set of objects satisfying the query (current result)
  - T- is the expiry time of the results
  - C- set of objects that will affect R at time T (<T,C> - TP component)

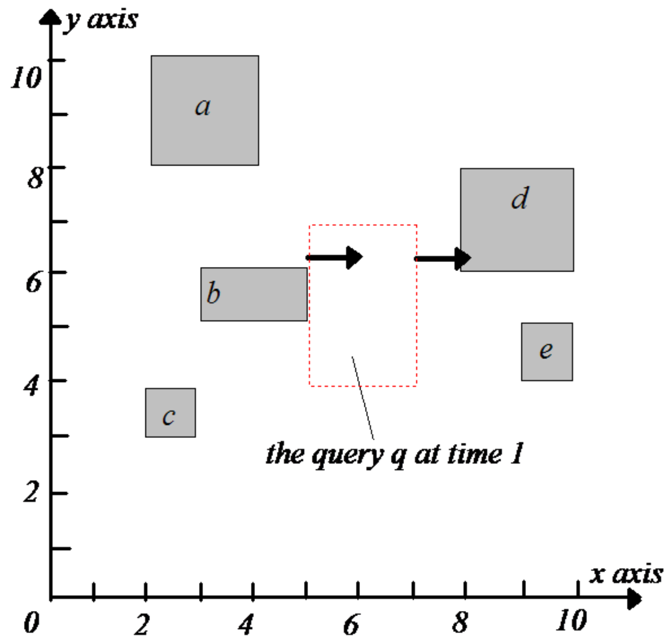




Time Parameterized (TP) queries

- Intro: Output form <R,T,C>
  - R- set of objects satisfying the query (current result)
  - T- is the expiry time of the results
  - C- set of objects that will affect R at time T (<T,C> - TP component)

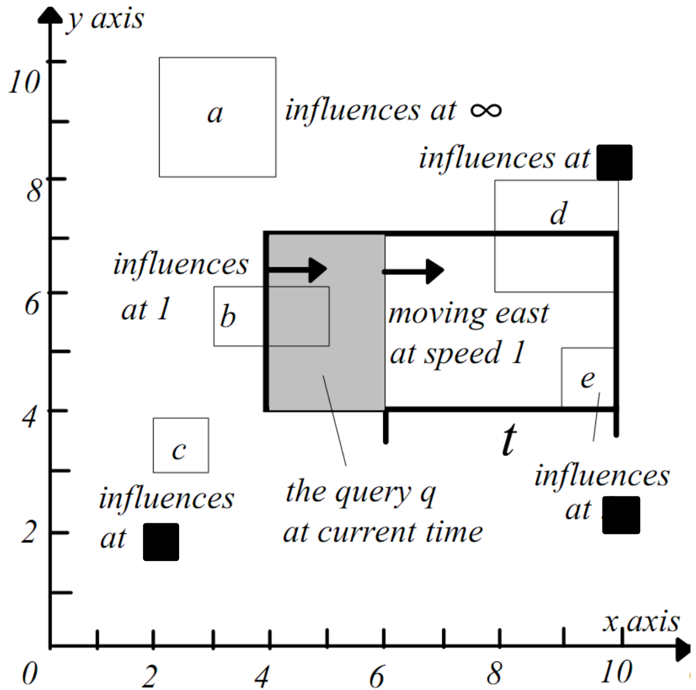




Result:<{b},1,{b}>

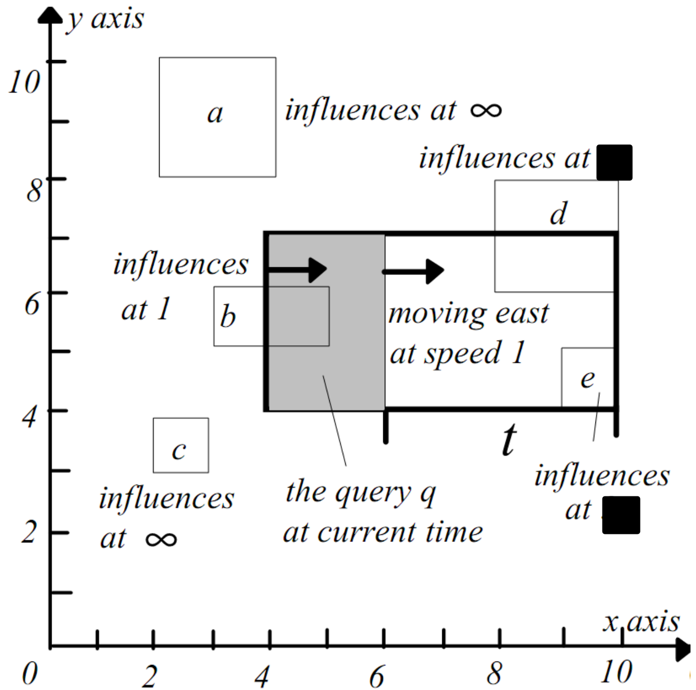
Time Parameterized (TP) queries

- Influence Time
  - Some objects influence the query at current time, but not in the future
  - Some objects are not currently in the result, but they may influence the query in the future.



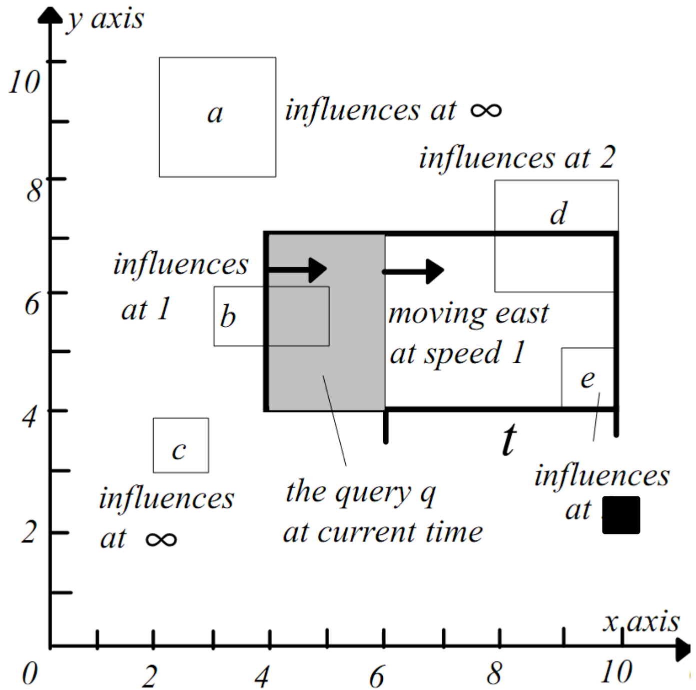
Time Parameterized (TP) queries

- Influence Time
  - Some objects influence the query at current time, but not in the future
  - Some objects are not currently in the result, but they may influence the query in the future.



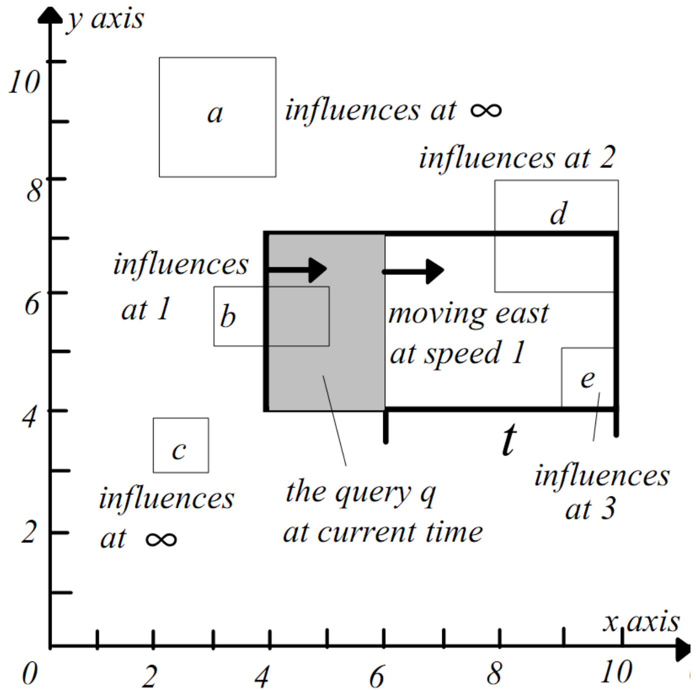
Time Parameterized (TP) queries

- Influence Time
  - Some objects influence the query at current time, but not in the future
  - Some objects are not currently in the result, but they may influence the query in the future.



Time Parameterized (TP) queries

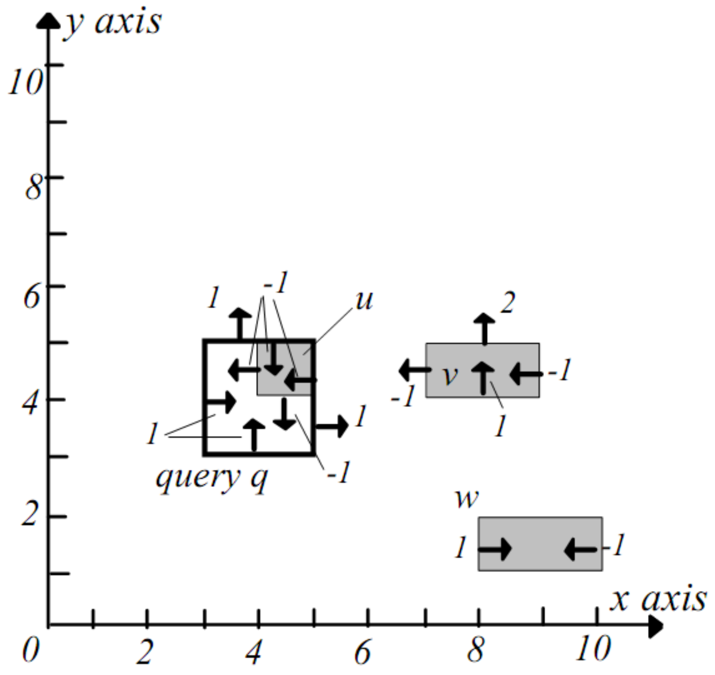
- Influence Time
  - Some objects influence the query at current time, but not in the future
  - Some objects are not currently in the result, but they may influence the query in the future.



TP window query

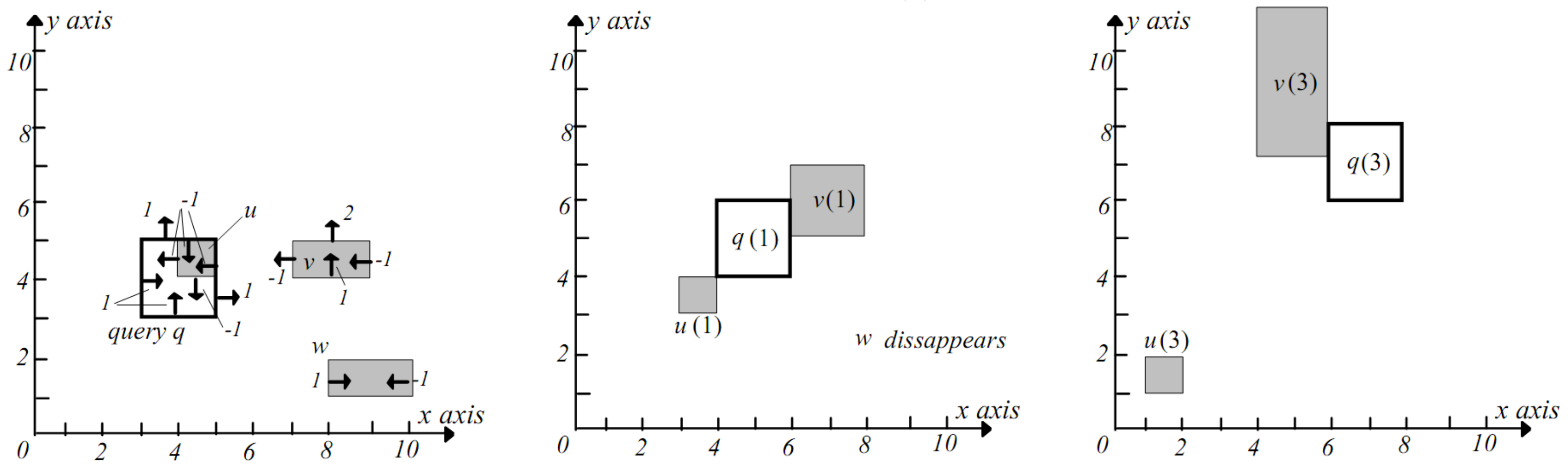
- To find the *influence time* $T_{INF}(o,q)$ of an object o, with the query window q, we need the *intersection period* $[T_s, T_e)$ during which o will intersect q.

- At time 0:



TP window query

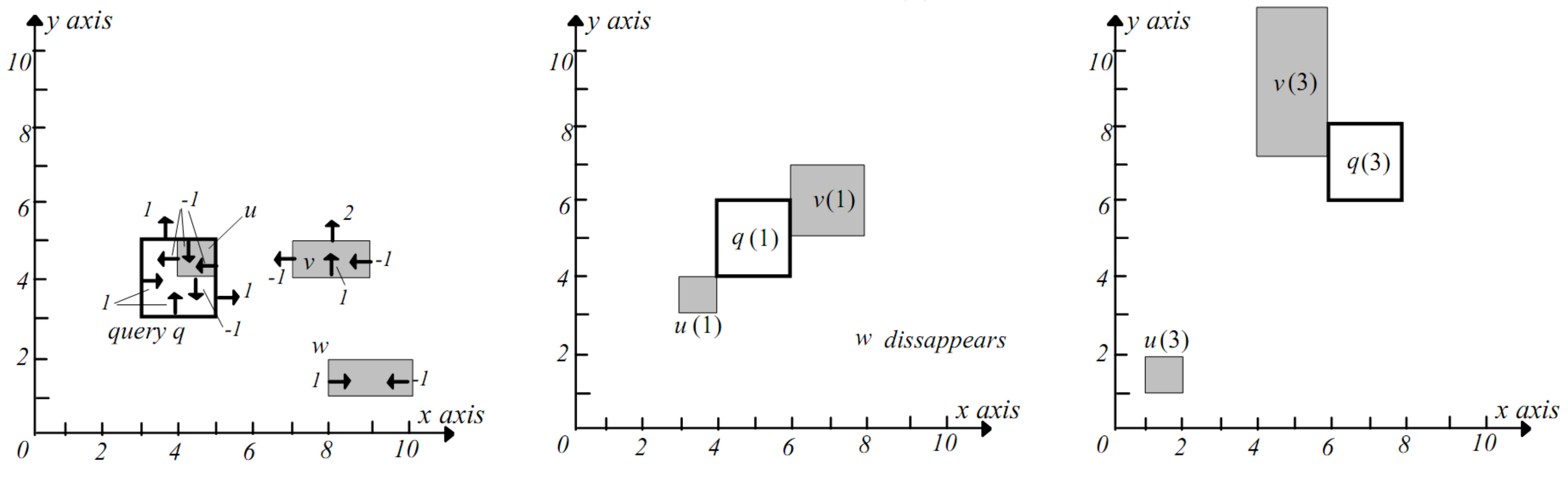
- What is the intersection time for u,v,w ?



(a) At time 0 (b) At time 1 (c) At time 3

TP window query

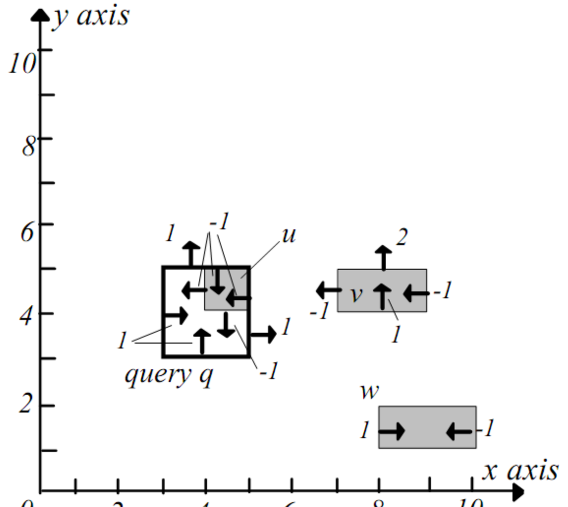
- What is the intersection time for u,v,w ?
  - u -> [0,1)



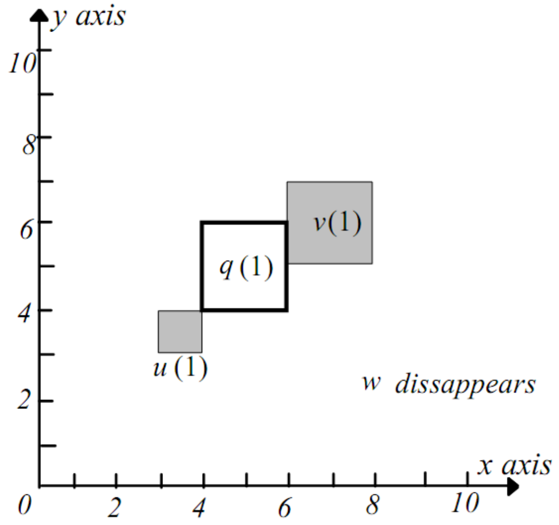
(a) At time 0 (b) At time 1 (c) At time 3

TP window query

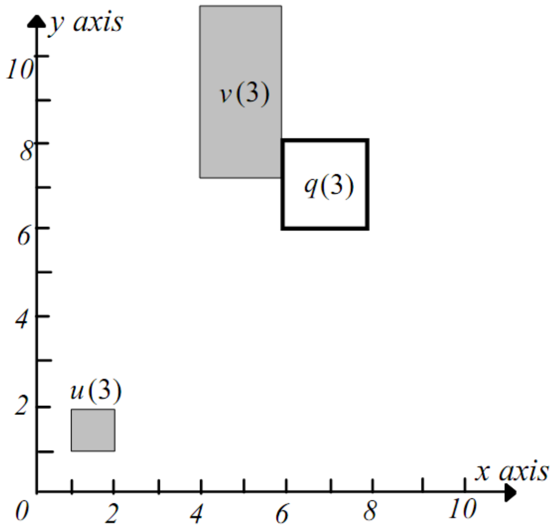
- What is the intersection time for u,v,w ?
  - u -> [0,1)
  - v->[1,3)



(a) At time 0



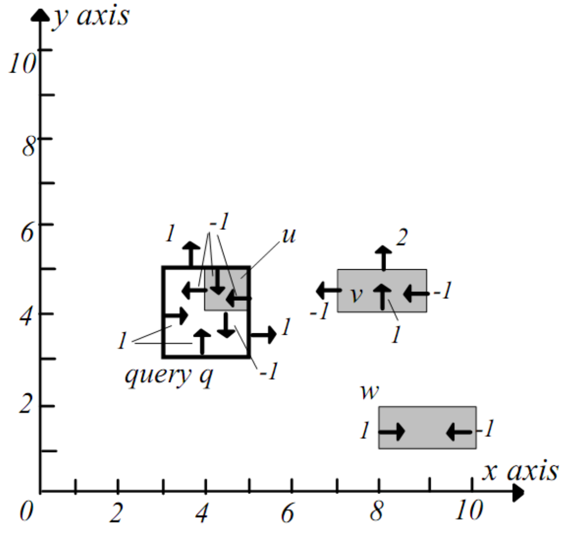
(b) At time 1



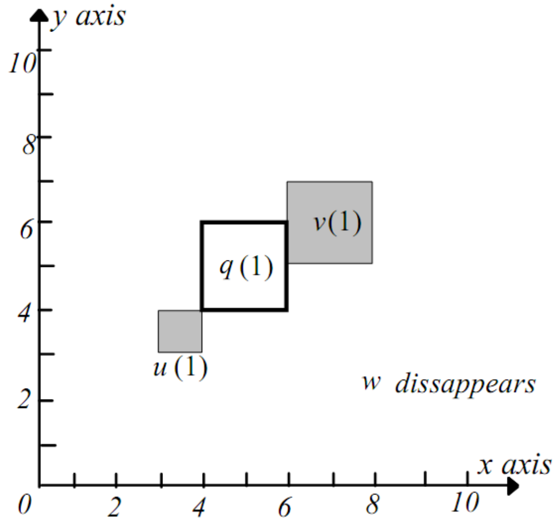
(c) At time 3

TP window query

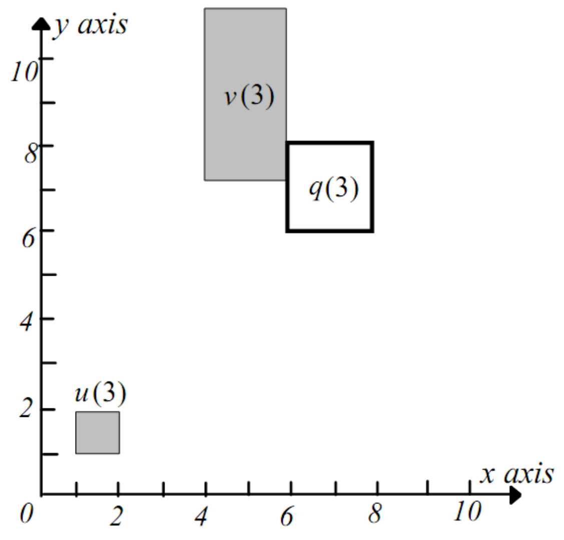
- What is the intersection time for u,v,w ?
  - u -> [0,1)
  - v->[1,3)
  - w->[inf, inf)



(a) At time 0



(b) At time 1



(c) At time 3

TP window query

TP window query

- For the i-th dimension:
  - Notation:
    - Object: MBR- [o_{iL}, o_{iR}] Velocity-[$o.V_{iL}$, $o.V_{iR}$]
    - Query: MBR- [q_{iL}, q_{iR}] Velocity-[$q.V_{iL}$, $q.V_{iR}$]

USC Viterbi
School of Engineering

TP window query

- For the i-th dimension:
  - Notation:
    - Object: MBR- [o_{iL}, o_{iR}] Velocity-[$o.V_{iL}$, $o.V_{iR}$]
    - Query: MBR- [q_{iL}, q_{iR}] Velocity-[$q.V_{iL}$, $q.V_{iR}$]

- Disappearance Time ($o.T_{iDSP}$)

TP window query

- For the i-th dimension:
  - Notation:
    - Object: MBR- [o_{iL}, o_{iR}] Velocity-[$o.V_{iL}$, $o.V_{iR}$]
    - Query: MBR- [q_{iL}, q_{iR}] Velocity-[$q.V_{iL}$, $q.V_{iR}$]

- Disappearance Time ($o.T_{iDSP}$)
  – $o.T_{iDSP} = (o_{iR} - o_{iL}) / (o.V_{iL} - o.V_{iR})$

TP window query

- For the i-th dimension:
  - Notation:
    - Object: MBR- [o_{iL}, o_{iR}] Velocity-[$o.V_{iL}$, $o.V_{iR}$]
    - Query: MBR- [q_{iL}, q_{iR}] Velocity-[$q.V_{iL}$, $q.V_{iR}$]

- Disappearance Time ($o.T_{iDSP}$)
  – $o.T_{iDSP} = (o_{iR} - o_{iL}) / (o.V_{iL} - o.V_{iR})$
  – The intersection between query and objects should happen before they disappear.

TP window query

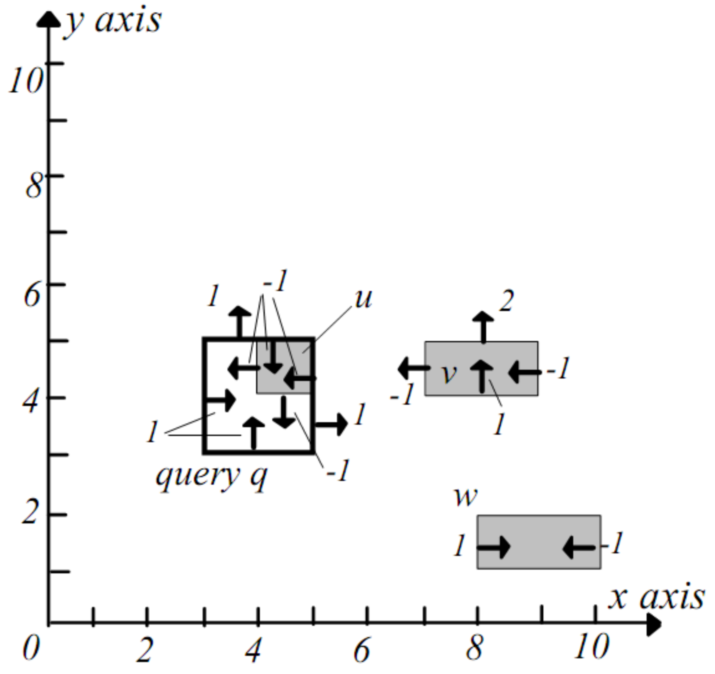
- For the i-th dimension:
  - Notation:
    - Object: MBR- [o_{iL}, o_{iR}] Velocity-[$o.V_{iL}$, $o.V_{iR}$]
    - Query: MBR- [q_{iL}, q_{iR}] Velocity-[$q.V_{iL}$, $q.V_{iR}$]

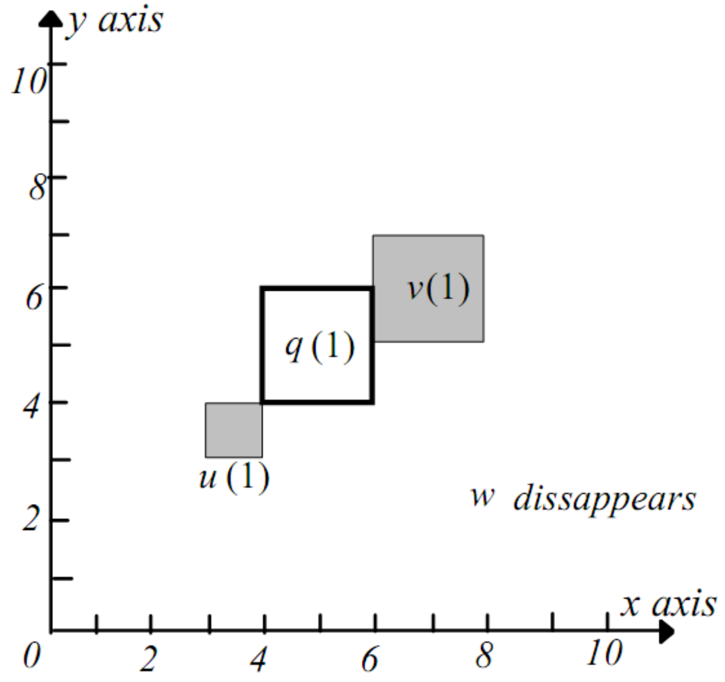
- Disappearance Time ($o.T_{iDSP}$)
  – $o.T_{iDSP} = (o_{iR} - o_{iL}) / (o.V_{iL} - o.V_{iR})$
  – The intersection between query and objects should happen before they disappear.
  – The influence time $T_{inf}(o,q)$ should be no later than
    - $Min(o.T_{DSP}, q.T_{DSP})$

USC Viterbi School of Engineering

TP window query

Example: Disappearance Time (o.T$_{iDSP}$) of *w*



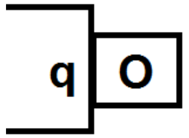


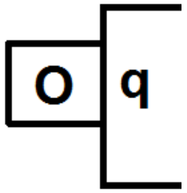
TP window query

- Core task:
  - Find intersects $[T_s, T_e)$ between object and query
  - Object o and query intersects if and only if they intersect along all dimensions

Computation: (for dimension i)

- If objects do not intersect at time 0, then:
  - T_{iLR}: Time the leftmost point of object meets the rightmost point of query
  - T_{iRL} : Time the rightmost point of object meets the leftmost point of query



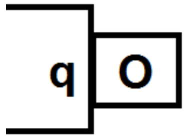


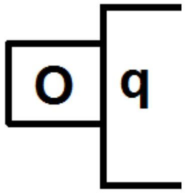
TP window query

- Core task:
  - Find intersects $[T_s, T_e)$ between object and query
  - Object o and query intersects if and only if they intersect along all dimensions

Computation: (for dimension i)

- If objects do not intersect at time 0, then:
  - T_{iLR}: Time the leftmost point of object meets the rightmost point of query
  - T_{iRL} : Time the rightmost point of object meets the leftmost point of query
  - $T_{is} = \min(T_{iLR}, T_{iRL})$ if neither disappears



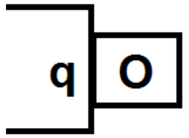


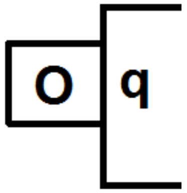
TP window query

- Core task:
  - Find intersects $[T_s, T_e)$ between object and query
  - Object o and query intersects if and only if they intersect along all dimensions

Computation: (for dimension i)

- If objects do not intersect at time 0, then:
  - T_{iLR}: Time the leftmost point of object meets the rightmost point of query
  - T_{iRL} : Time the rightmost point of object meets the leftmost point of query
  - $T_{is} = \min(T_{iLR}, T_{iRL})$ if neither disappears
  - $T_{ie} = \min(\max(T_{iLR}, T_{iRL}), o.T_{DSP}, q.T_{DSP})$



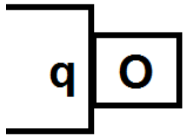


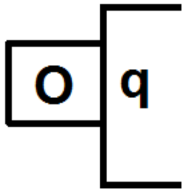
TP window query

- Core task:
  - Find intersects $[T_s, T_e)$ between object and query
  - Object o and query intersects if and only if they intersect along all dimensions

Computation: (for dimension i)

- If objects do not intersect at time 0, then:
  - T_{iLR}: Time the leftmost point of object meets the rightmost point of query
  - T_{iRL} : Time the rightmost point of object meets the leftmost point of query
  - $T_{is} = \min(T_{iLR}, T_{iRL})$ if neither disappears
  - $T_{ie} = \min(\max(T_{iLR}, T_{iRL}), o.T_{DSP}, q.T_{DSP})$
- If objects already intersect, then $T_{is} = 0$, and $T_{ie} = \min(T_{iLR}, T_{iRL}, o.T_{DSP}, q.T_{DSP})$



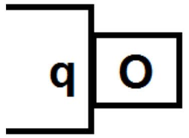


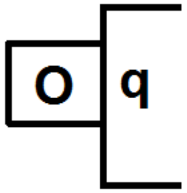
TP window query

- Core task:
  - Find intersects $[T_s, T_e)$ between object and query
  - Object o and query intersects if and only if they intersect along all dimensions

Computation: (for dimension i)

- If objects do not intersect at time 0, then:
  - T_{iLR}: Time the leftmost point of object meets the rightmost point of query
  - T_{iRL} : Time the rightmost point of object meets the leftmost point of query
  - $T_{is} = \min(T_{iLR}, T_{iRL})$ if neither disappears
  - $T_{ie} = \min(\max(T_{iLR}, T_{iRL}), o.T_{DSP}, q.T_{DSP})$
- If objects already intersect, then $T_{is} = 0$, and $T_{ie}=\min(T_{iLR}, T_{iRL}, o.T_{DSP}, q.T_{DSP})$
- $[T_s, T_e) = \cap[T_{is}, T_{ie})$





TP window query

- Important Notation:
  - Intersection period:
    $[T_s, T_e)$: the time that object and query intersect
  - Influence time T_{INF} (o,q):

TP window query

- Important Notation:
  - Intersection period:
    $[T_s, T_e)$: the time that object and query intersect
  - Influence time $T_{INF}(o,q)$:
    - If o does not currently intersect the query
      - $T_{INF}(o,q) = T_s$

USC Viterbi School of Engineering

TP window query

- Important Notation:
  - Intersection period:
    $[T_s, T_e)$: the time that object and query intersect
  - Influence time $T_{INF}(o,q)$:
    - If o does not currently intersect the query
      - $T_{INF}(o,q) = T_s$
    - If o is currently intersecting the query
      - $T_{INF}(o,q) = T_e$

TP window query

- Important Notation:
  - Intersection period:
    $[T_s, T_e)$: the time that object and query intersect
  - Influence time $T_{INF}(o,q)$:
    - If o does not currently intersect the query
      - $T_{INF}(o,q) = T_s$
    - If o is currently intersecting the query
      - $T_{INF}(o,q) = T_e$
  - The expiry time of the current result is the minimum influence time of all objects

Window Query processing

Window Query processing

- Both Best-FS and DFS can be used for processing TP queries.

Window Query processing

- Both Best-FS and DFS can be used for processing TP queries.
- Algorithm:
  - Start with R = Null, T = infinity, C = Null

Window Query processing

- Both Best-FS and DFS can be used for processing TP queries.
- Algorithm:
  - Start with R = Null, T = infinity, C = Null
  - For each object o

Window Query processing

- Both Best-FS and DFS can be used for processing TP queries.
- Algorithm:
  - Start with R = Null, T = infinity, C = Null
  - For each object o
    - If o satisfy q, then R = R U{o}

USC Viterbi
School of Engineering

Window Query processing

- Both Best-FS and DFS can be used for processing TP queries.
- Algorithm:
  - Start with R = Null, T = infinity, C = Null
  - For each object o
    - If o satisfy q, then R = R U{o}

    - If $T_{INF}(o,q) < T$
      - C={o}
      - $T = T_{INF}(o,q)$

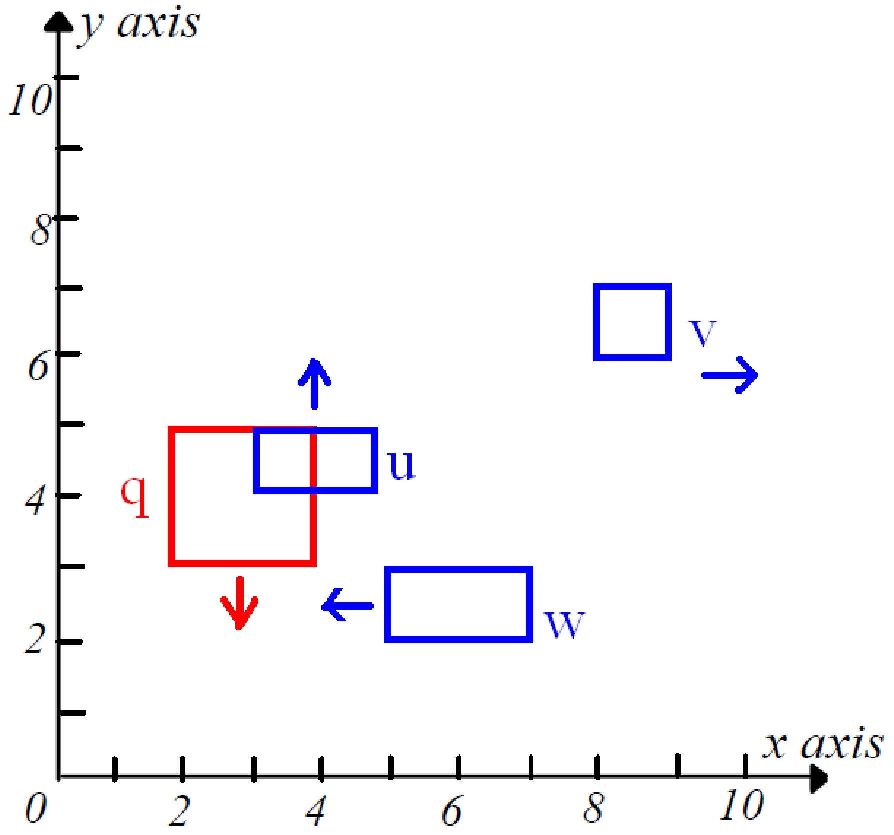
Window Query processing

- Both Best-FS and DFS can be used for processing TP queries.
- Algorithm:
  - Start with R = Null, T = infinity, C = Null
  - For each object o
    - If o satisfy q, then R = R U{o}

    - If $T_{INF}(o,q) < T$
      - C={o}
      - $T = T_{INF}(o,q)$
    - Else if $T_{INF}(o,q) = T$
      - C = C U{o}

Example:

- Assuming their velocity is 1.



Intersect time:

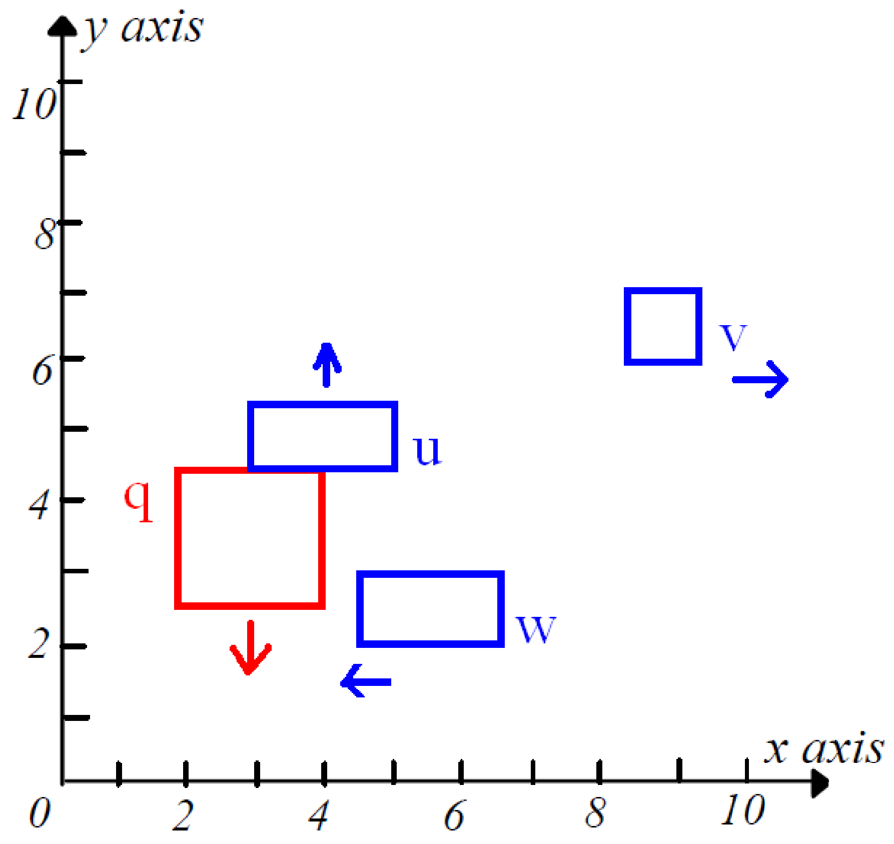
u: [0,0.5)
v: [inf, inf)
w: [1,3)

Query Result for time [0,3) ?

t = 0

Example:

- Assuming their velocity is 1.



Intersect time:

u: [0,0.5)
v: [inf, inf)
w: [1,3)

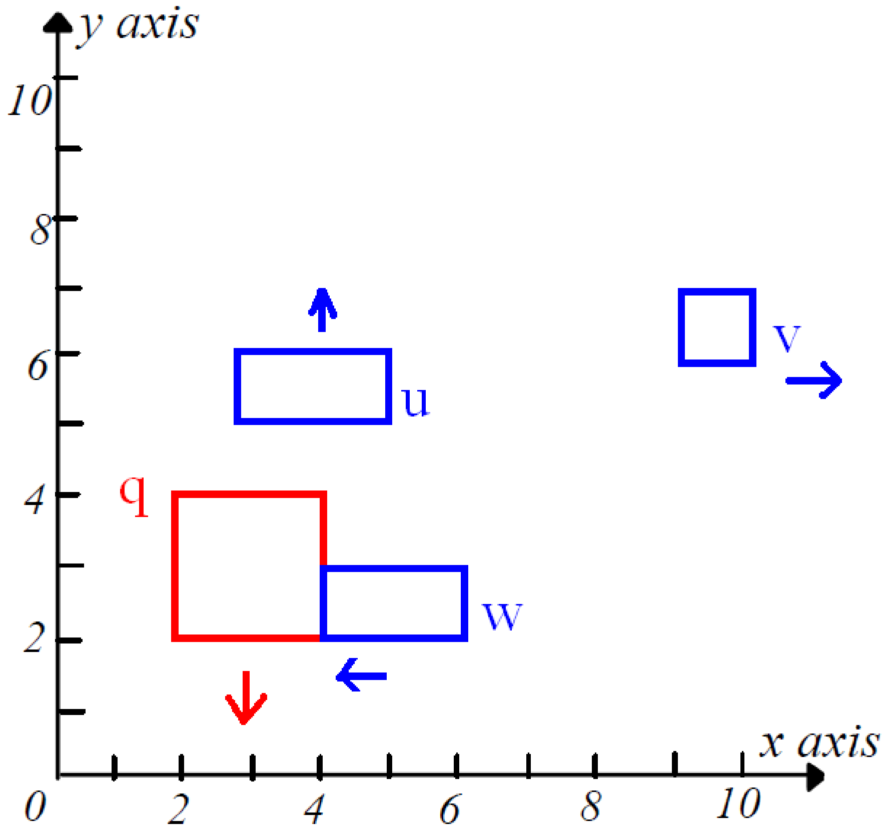
Query Result for time [0,3) ?

<{u}, 0.5, {u}>

t = 0.5

Example:

- Assuming their velocity is 1.



Intersect time:

u: [0,0.5)
v: [inf, inf)
w: [1,3)

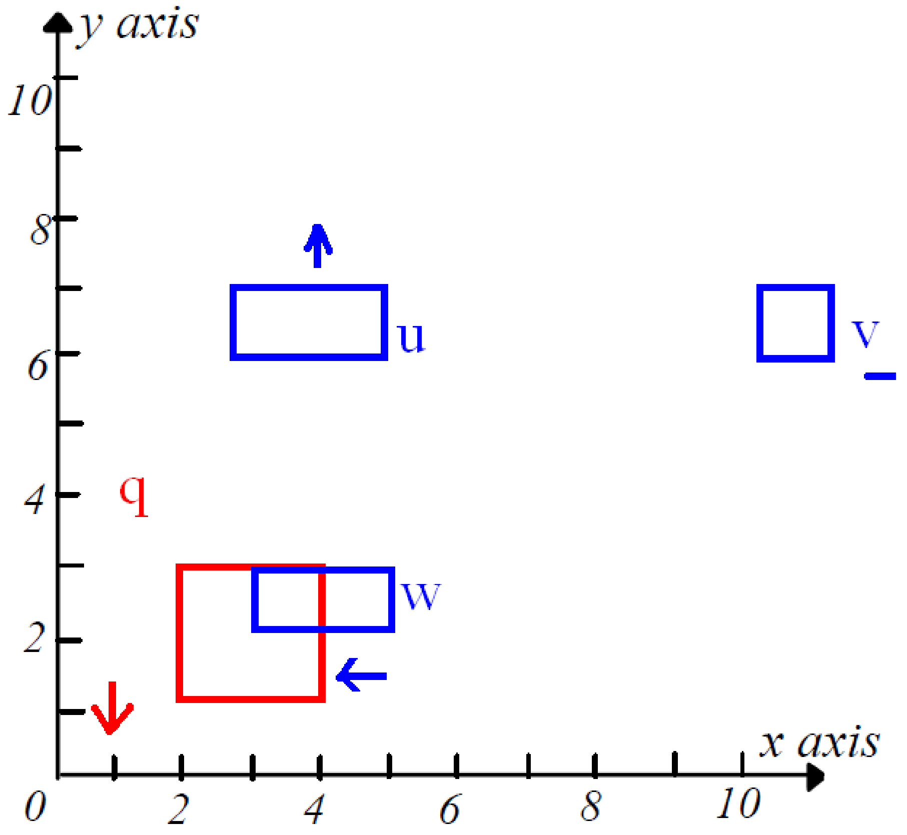
Query Result for time [0,3) ?

<{u}, 0.5, {u}>
<{}, 1, {w}>

t = 1

Example:

- Assuming their velocity is 1.



Intersect time:

u: [0,0.5)
v: [inf, inf)
w: [1,3)

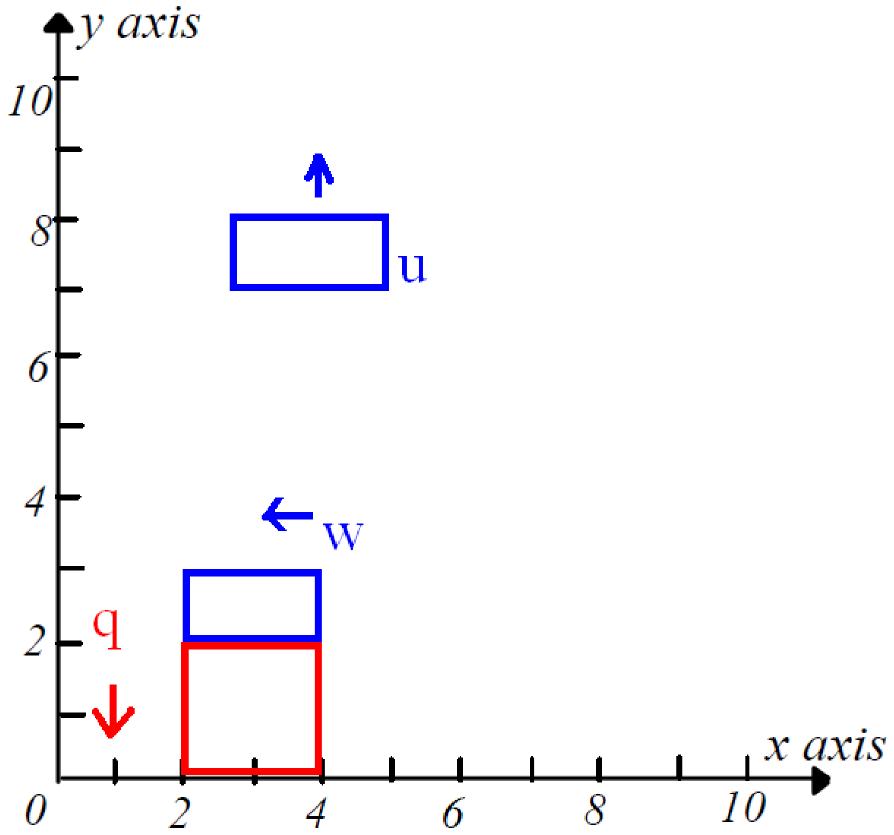
Query Result for time [0,3) ?

<{u}, 0.5, {u}>
<{}, 1, {w}>

t = 2

Example:

- Assuming their velocity is 1.



Intersect time:

u: [0,0.5)
v: [inf, inf)
w: [1,3)

Query Result for time [0,3) ?

<{u}, 0.5, {u}>
<{}, 1, {w}>
<{w},3, {w}>

t = 3

TP window query – Intermediate Node

TP window query – Intermediate Node

- Influence time of the intermediate Entry E corresponds to the minimum possible influence time of any object in the subtree of E

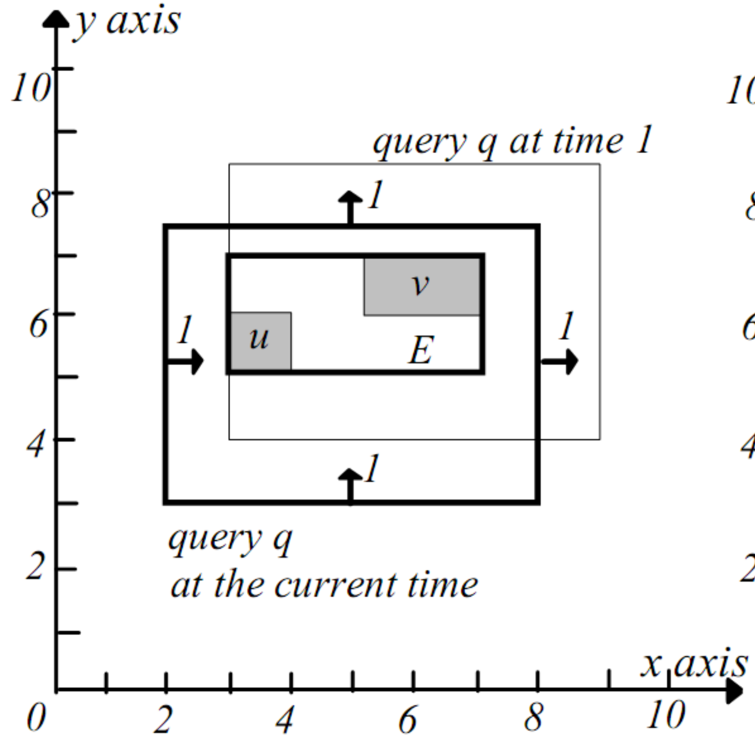
USC Viterbi School of Engineering

TP window query – Intermediate Node

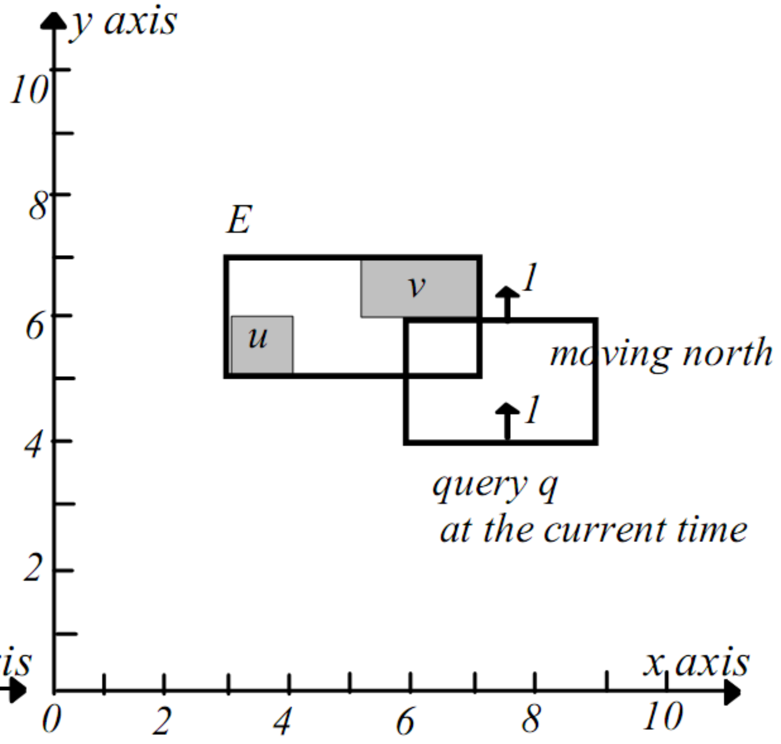
- Influence time of the intermediate Entry E corresponds to the minimum possible influence time of any object in the subtree of E
- If E does not intersect with q, then $T_{INF}(E,q)$ is the time E starts intersecting with q – because this is also the earliest time when any of the objects inside E can intersect (influence) q

TP window query -- Intermediate Node

- If E intersects q, then two cases:



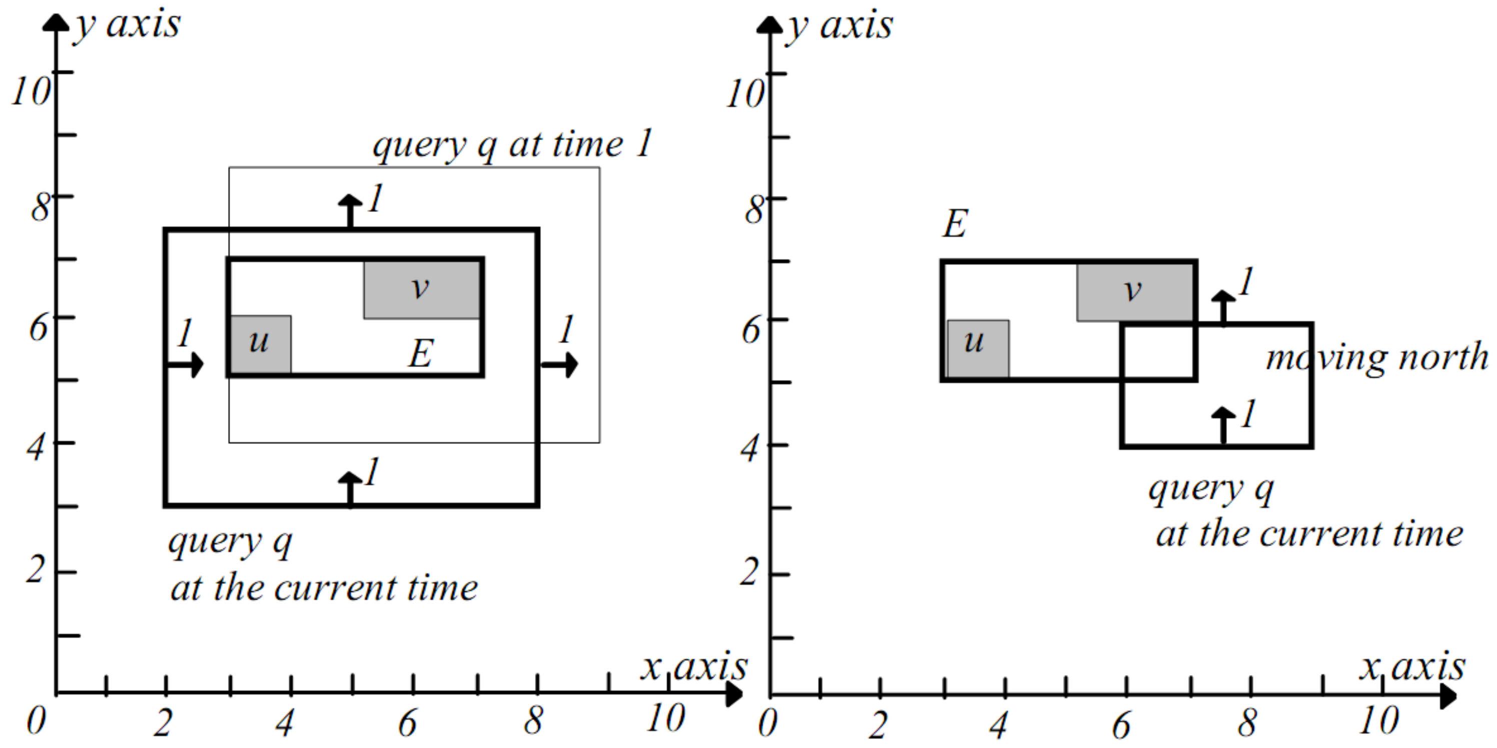
(a) *E* is contained in *q*



(b) *E* partially intersects *q*

TP window query -- Intermediate Node

- If E intersects q, then two cases:

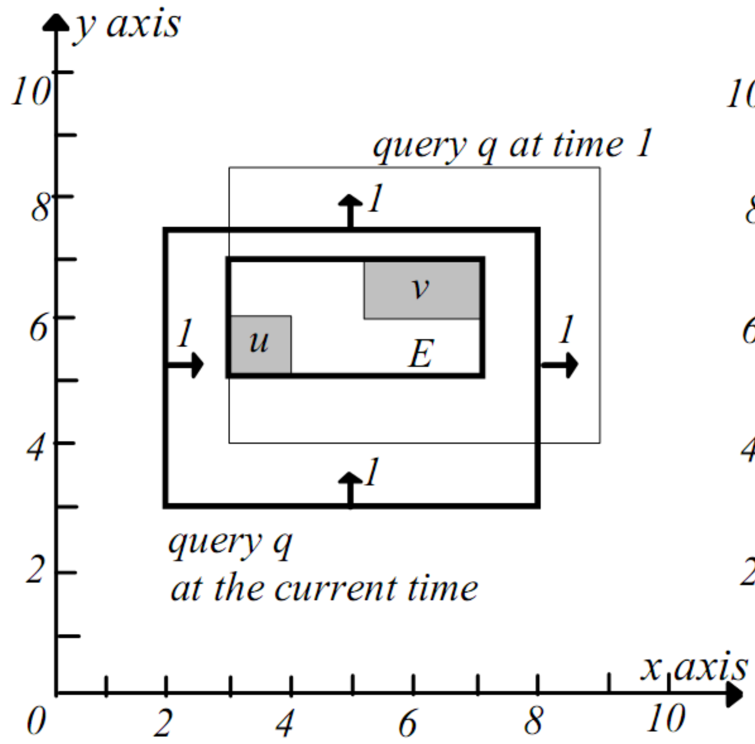


(a) *E* is contained in *q* (b) *E* partially intersects *q*

T_{INF}(E,q) =the time that E starts to partially intersect q in the Future (and at lease one of its object may stop intersecting soon)

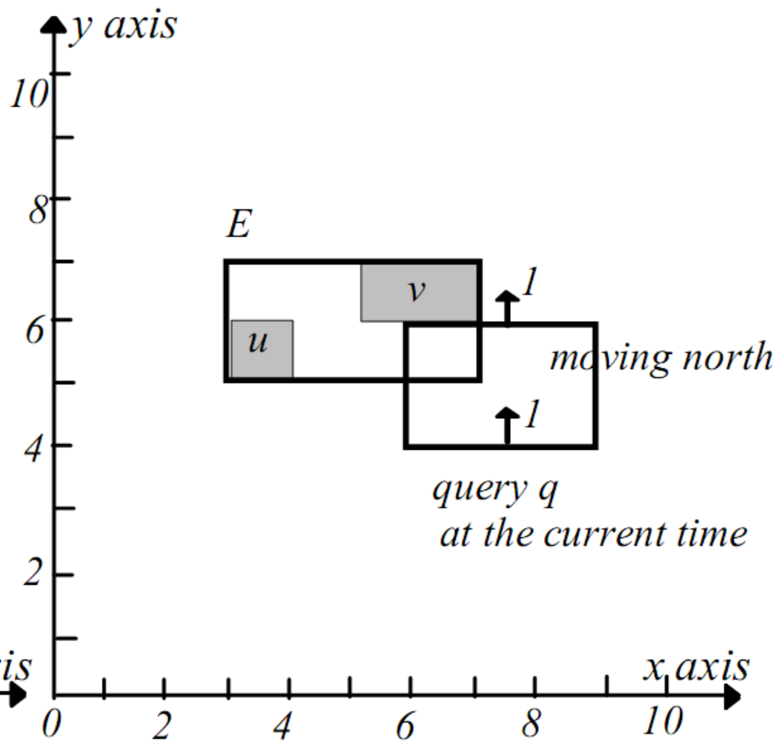
TP window query -- Intermediate Node

- If E intersects q, then two cases:



(a) *E* is contained in *q*

T_{INF}(E,q) =the time that E starts to partially intersect q in the Future (and at lease one of its object may stop intersecting soon)



(b) *E* partially intersects *q*

T_{INF}(E,q) = 0, because some object inside E (e.g., v) may influence the result as soon as the query move

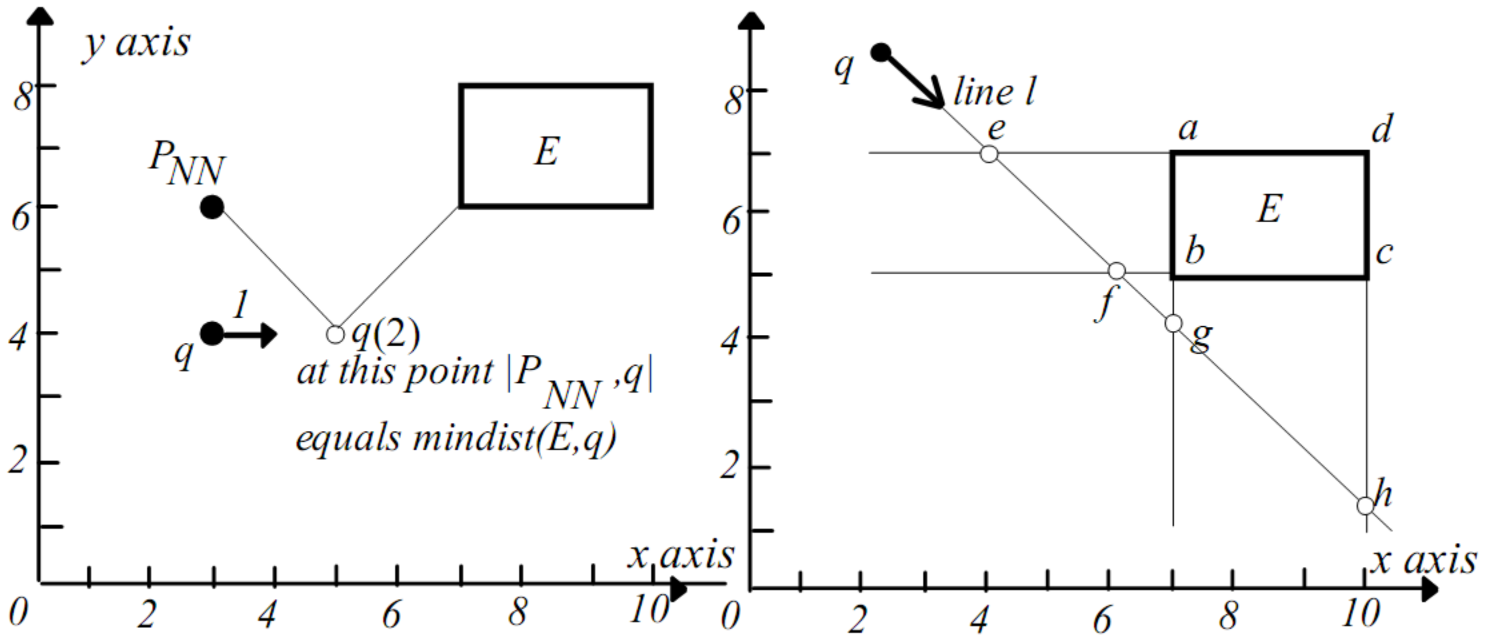
TP NN query

- Start with single NN
  - Influence time is no longer intersection period.
  - Instead, it is the time the object becomes the NN result.

  - T_{INF} is the minimum t that satisfies:
    - $Dist(o(t), q(t)) < Dist(P_{NN}(t), q(t))$
      - P_{NN} is the current NN of q.

TP NN query

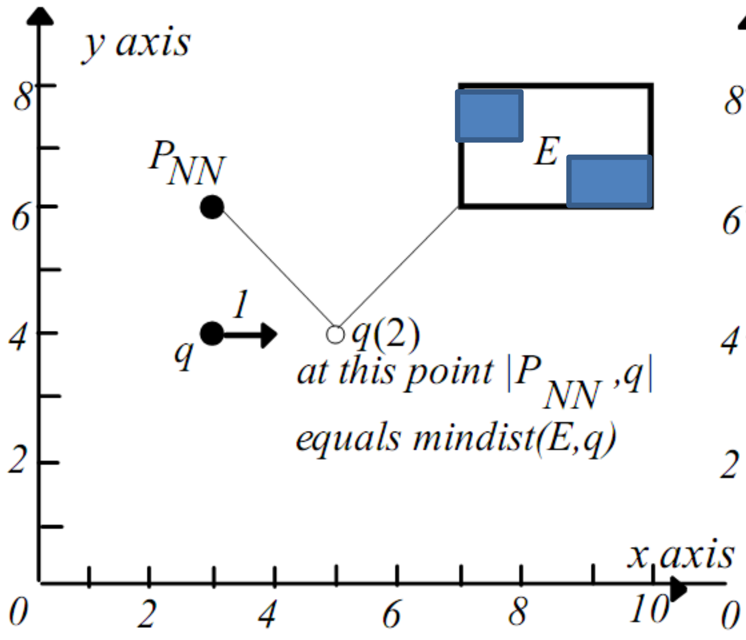
- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E



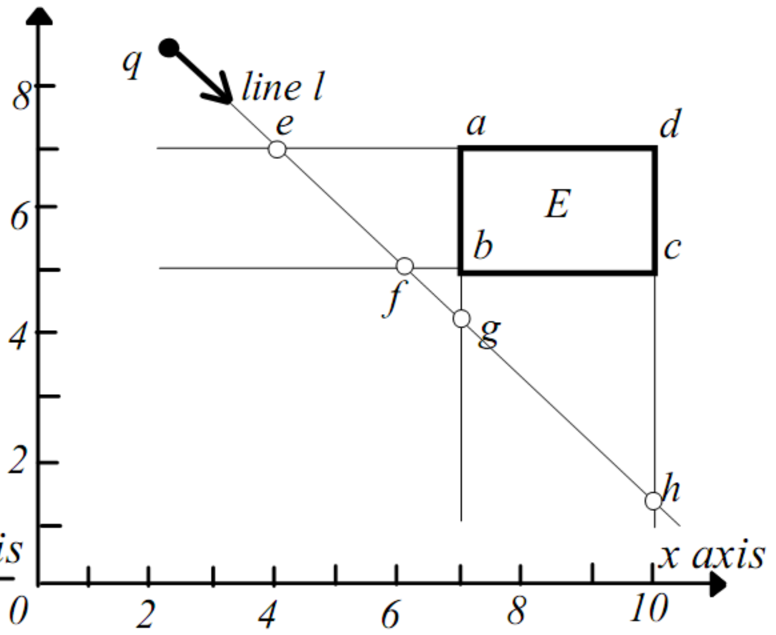
(a) Example of $T_{INF}(E,q)$ (b) Different cases of mindist

TP NN query

- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E



(a) Example of $T_{INF}(E,q)$

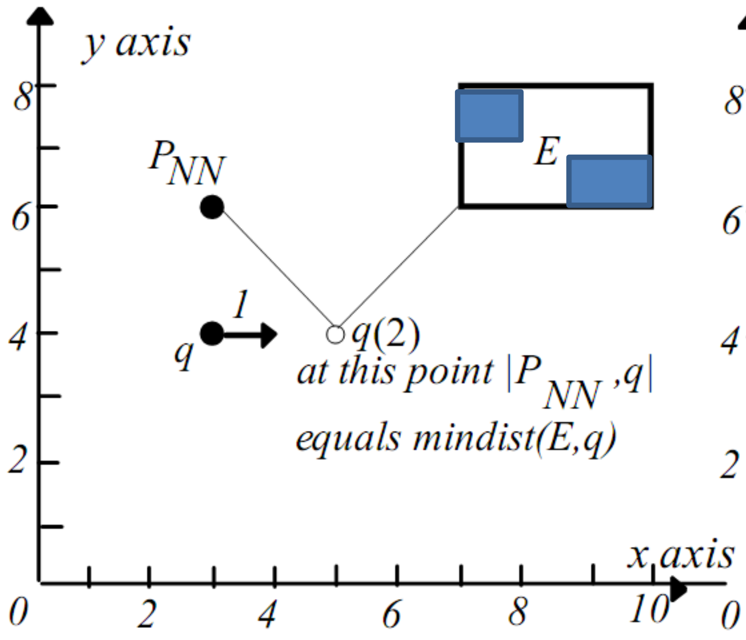


(b) Different cases of mindist

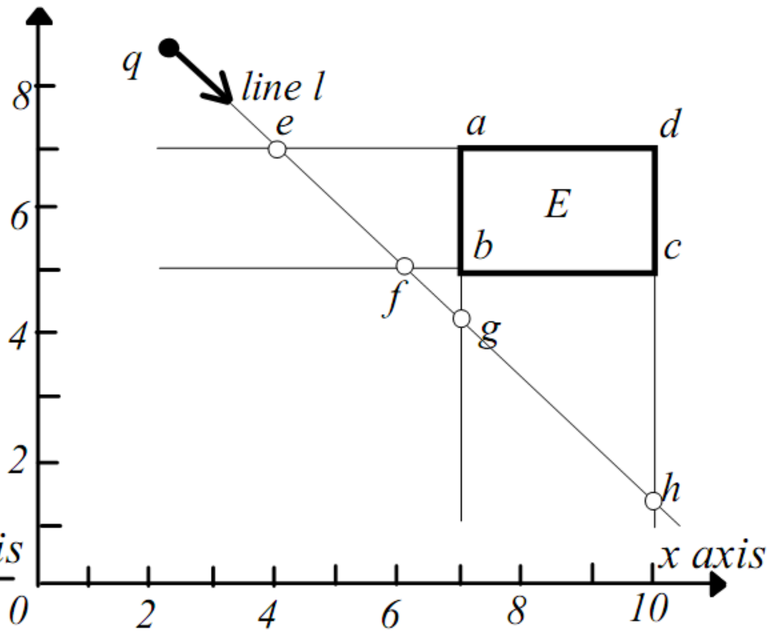
USC Viterbi School of Engineering

TP NN query

- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E



(a) Example of $T_{INF}(E,q)$



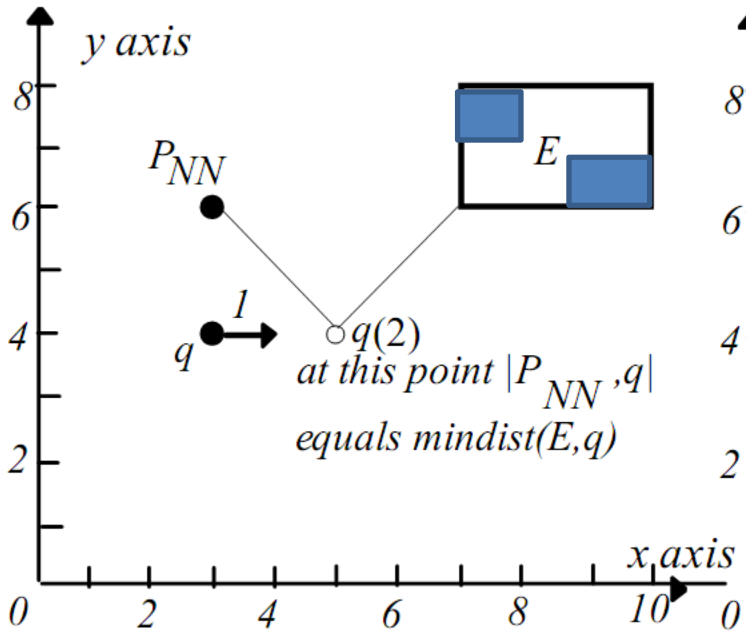
(b) Different cases of mindist

The computation of *mindist(E(t),q(t))* depends on the relative positions of E and q.

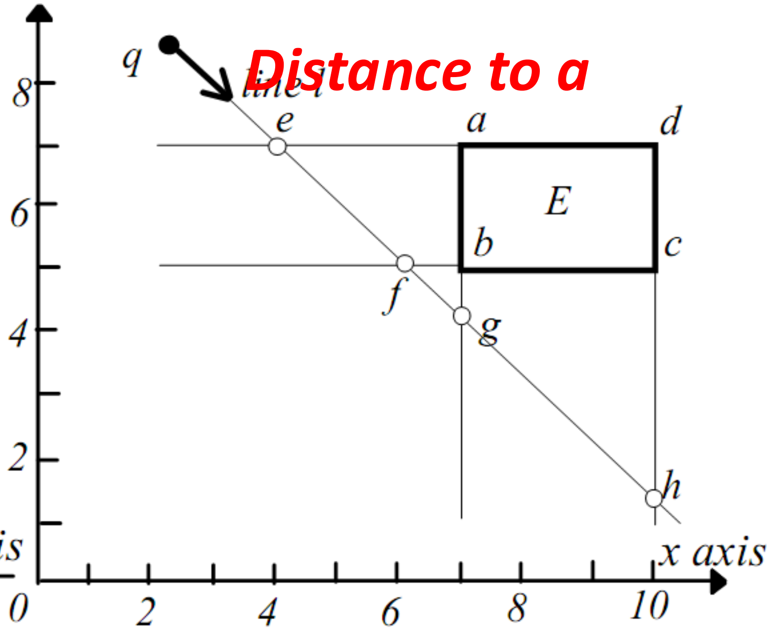
USC Viterbi School of Engineering

TP NN query

- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E



(a) Example of $T_{INF}(E,q)$

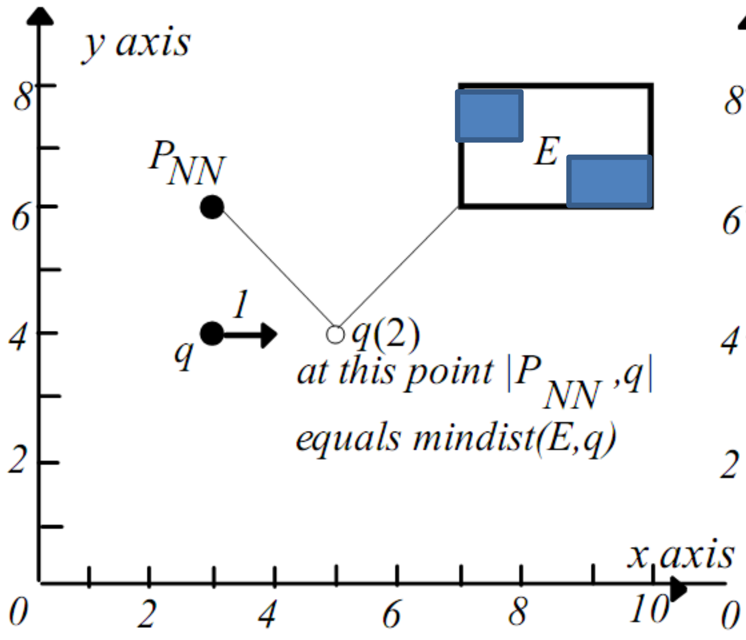


(b) Different cases of mindist

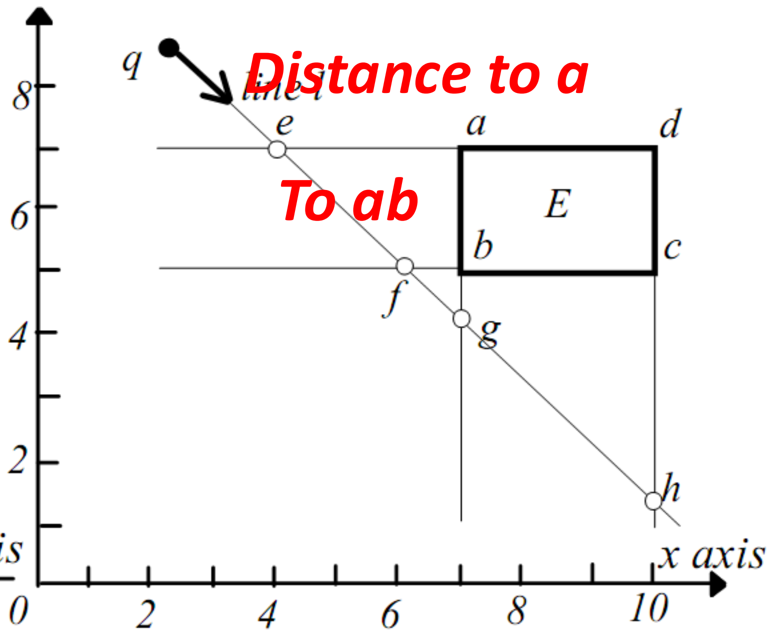
The computation of *mindist(E(t),q(t))* depends on the relative positions of E and q.

TP NN query

- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E



(a) Example of $T_{INF}(E,q)$

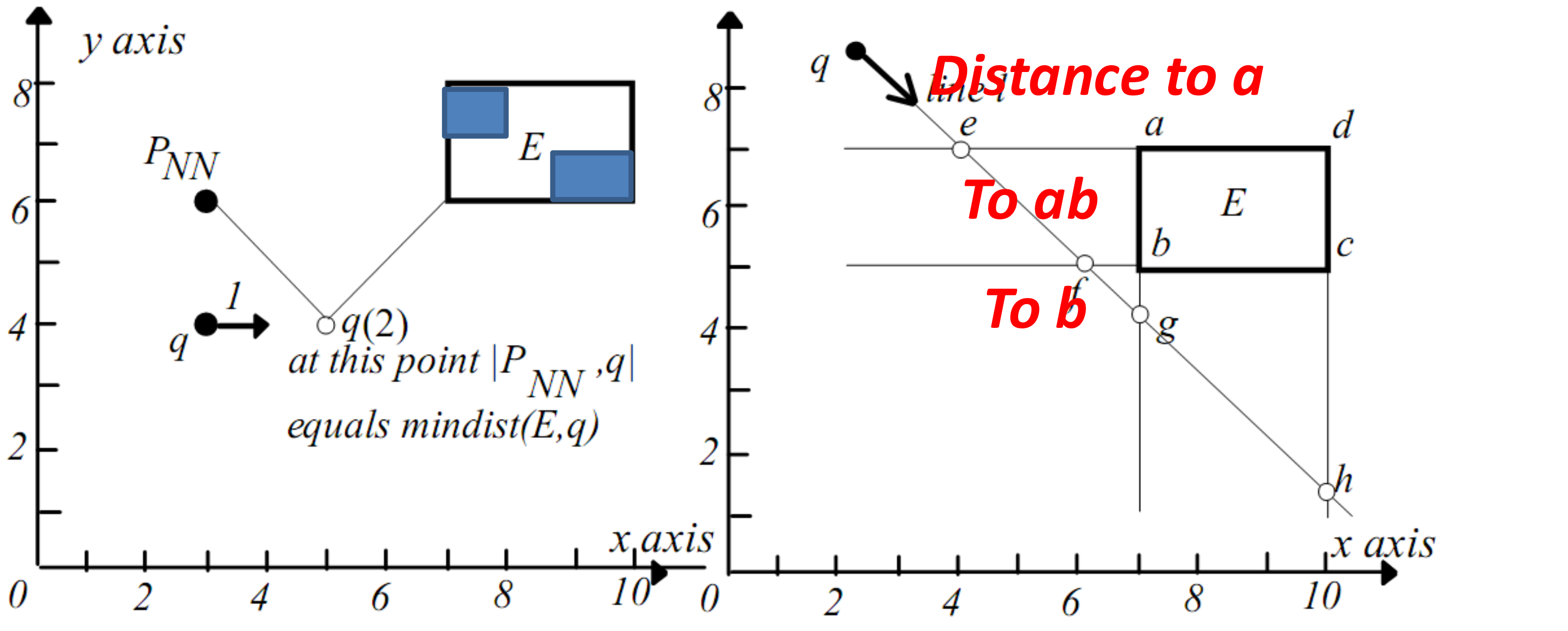


(b) Different cases of mindist

The computation of *mindist(E(t),q(t))* depends on the relative positions of E and q.

TP NN query

- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E

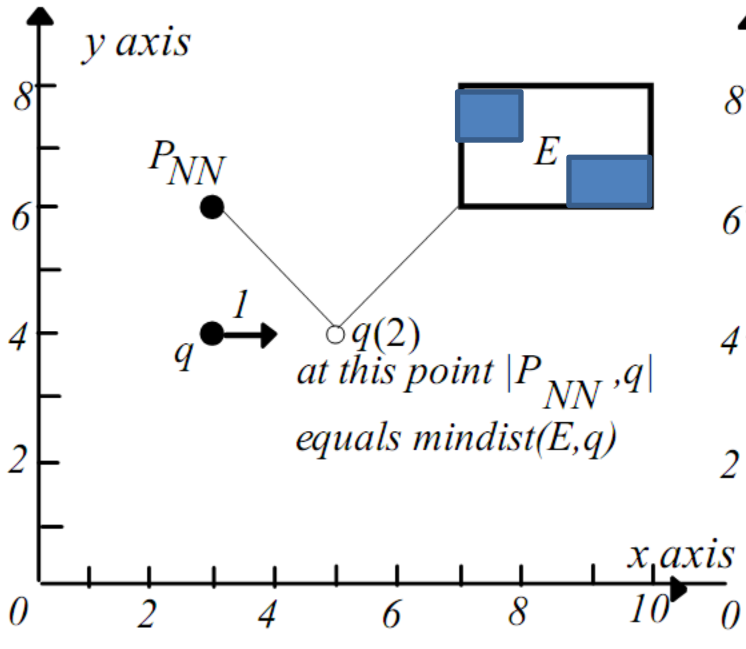


(a) Example of $T_{INF}(E,q)$ (b) Different cases of mindist

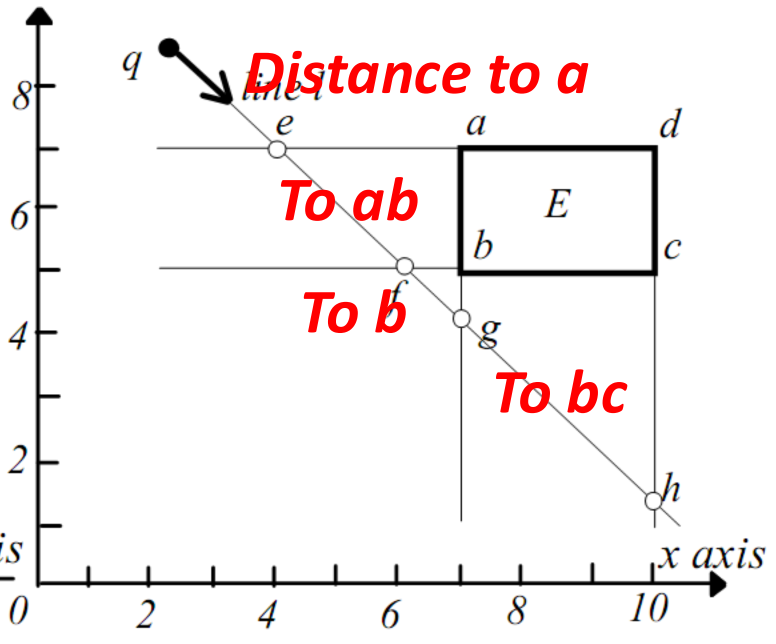
The computation of *mindist(E(t),q(t))* depends on the relative positions of E and q.

TP NN query

- Influence time of the Intermediate Entry E
  - Use mindist as the dist between q and E



(a) Example of $T_{INF}(E,q)$

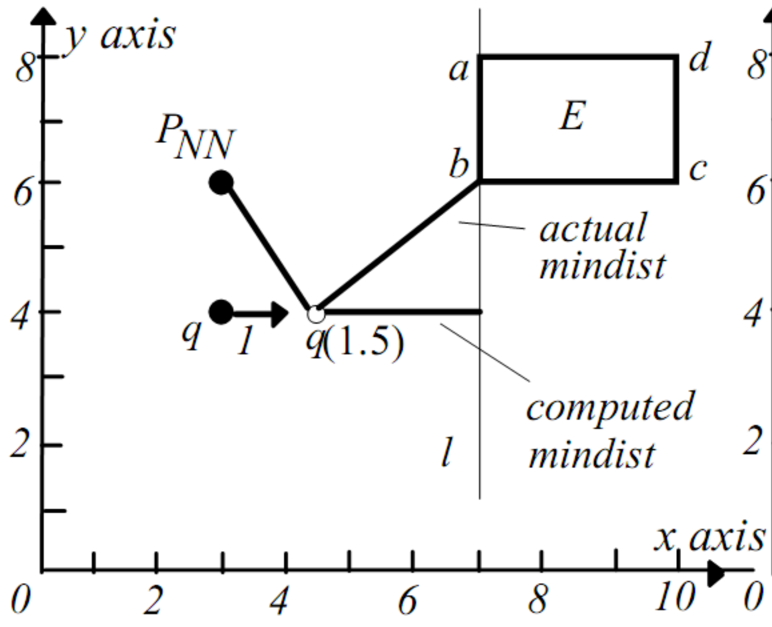


(b) Different cases of mindist

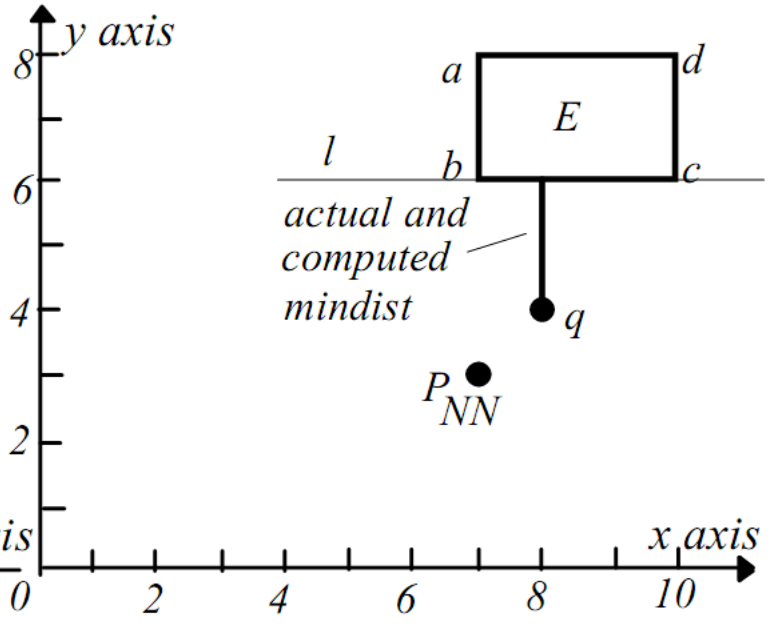
The computation of *mindist(E(t),q(t))* depends on the relative positions of E and q.

TP NN query

- Simpler Calculation (underestimates *mindist* (to ensure the correctness of BaB algorithms).



If mindist to a point, then distance to farther edge (ab) containing the point



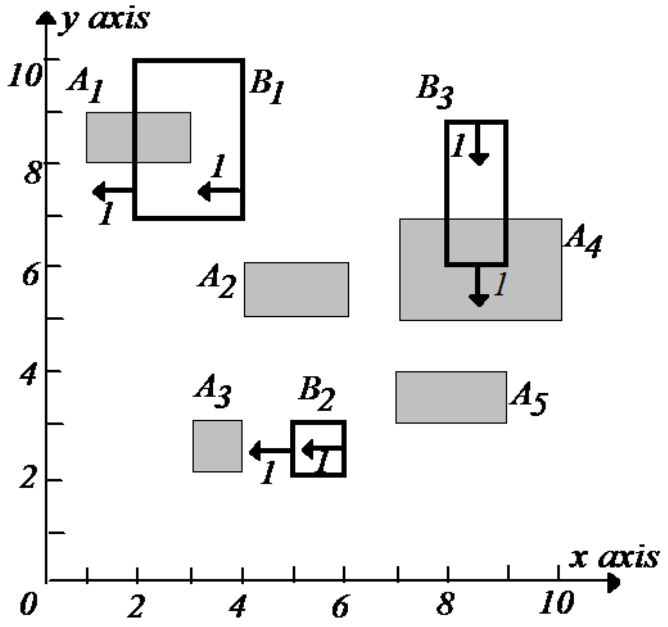
If mindist to an edge, then mindist at current time

TP join query

- A Join query returns all pairs of objects from two datsets that satisfy some spatial condition. (e.g., Intersection)
- A join result can change in the future when,
  - A pair of objects in the current result ceases to satisfy the join condition
  - New objects start satisfying the join condition
- TP join can be regarded as a closest pair query by treating $T_{INF}(o1,o2)$, instead of $T_{INF}(q,o)$

TP join query

- Influence time of object pairs



	A_1	A_2	A_3	A_4	A_5
B_1	3	∞	∞	∞	∞
B_2	∞	∞	1	∞	∞
B_3	∞	∞	∞	4	2

- *The* expiry time is the minimum influence time (i.e., $T_{INF}(A_3,B_2)$=l).

Conclusion

- Introduction of the novel concept of time-parameterized queries.
- Techniques for transforming the most common spatial queries to their TP counterparts.
- Development of efficient processing methods.

- Drawbacks?

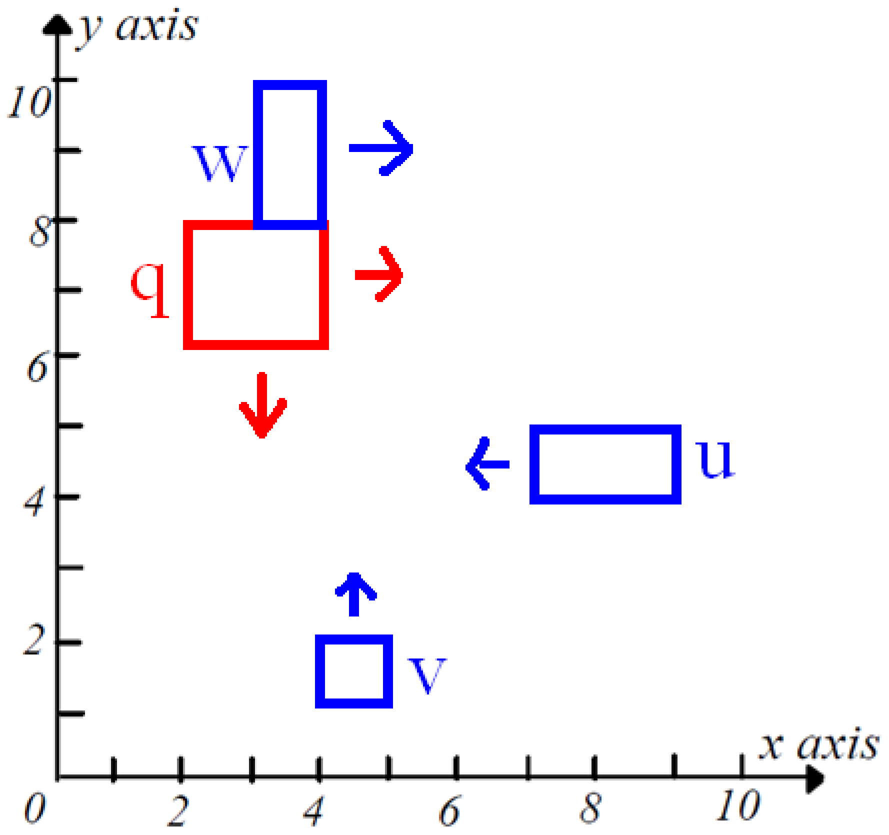
References

- Tao, Y. & Papadias, D. Time-parameterized queries in spatio-temporal databases . SIGMOD Conference, 2002, 334-345.
- A presentation by Penny Pan in csci587 Fall'2010

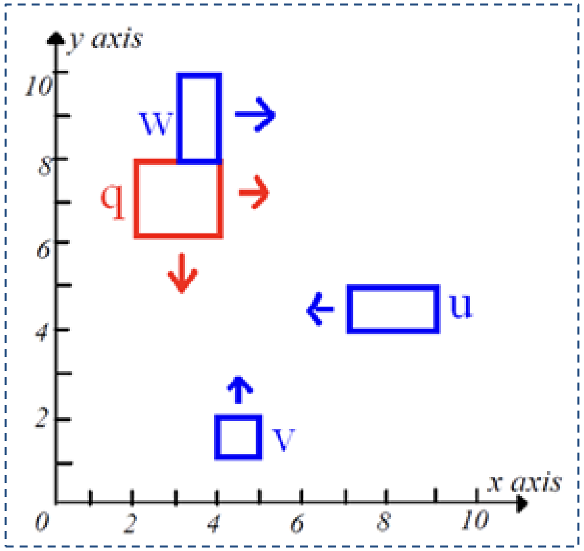
USC Viterbi
School of Engineering

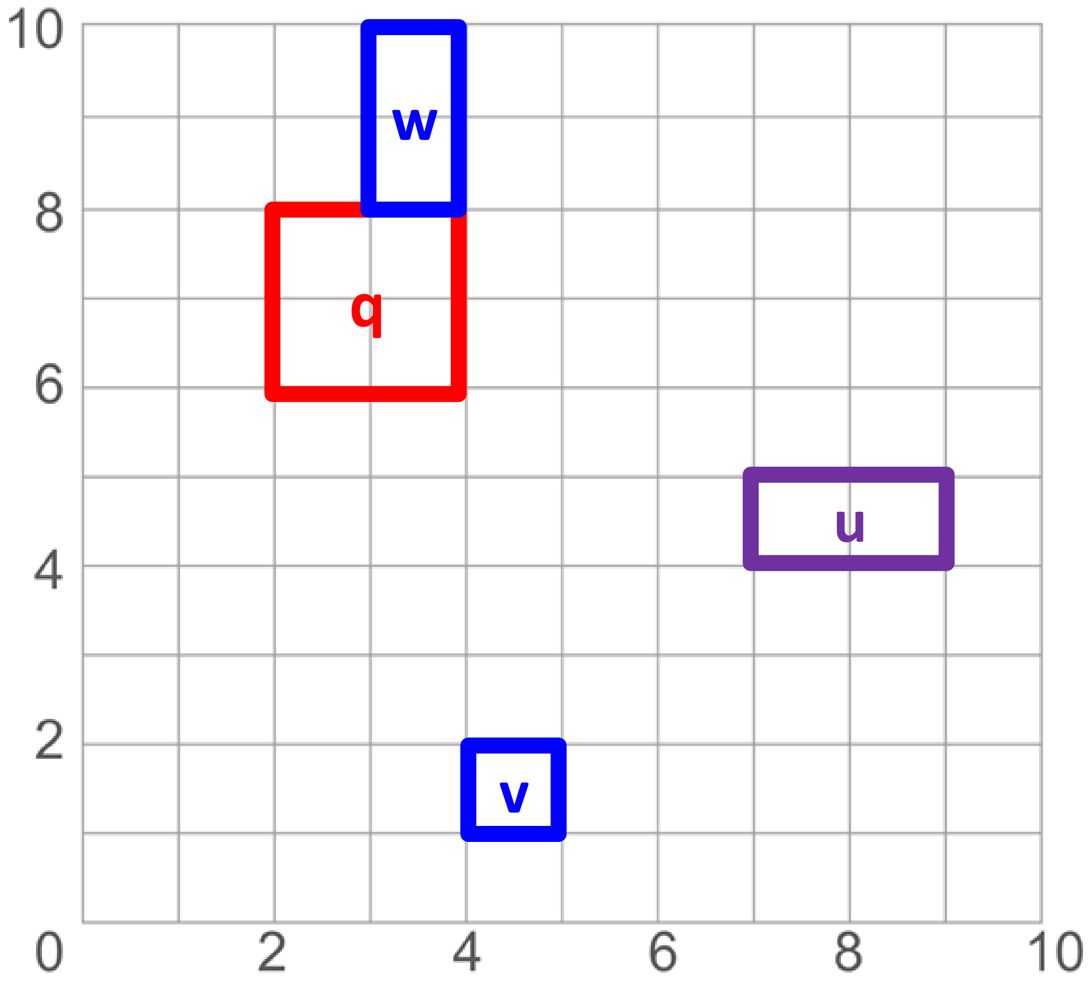
Practice

- Queries / objects are aligned with integer coordinates
  - All arrows mean 1 grid/time velocity
- Questions:
  - What is the Influence time of all objects?
  - What is the query result using TP strategy during time [0,3.5)?

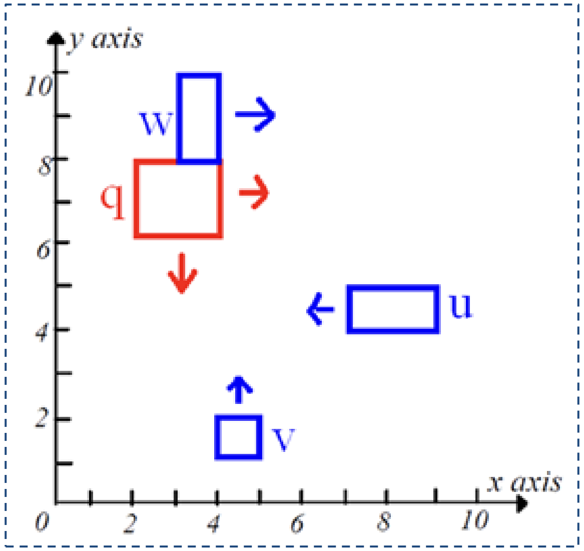


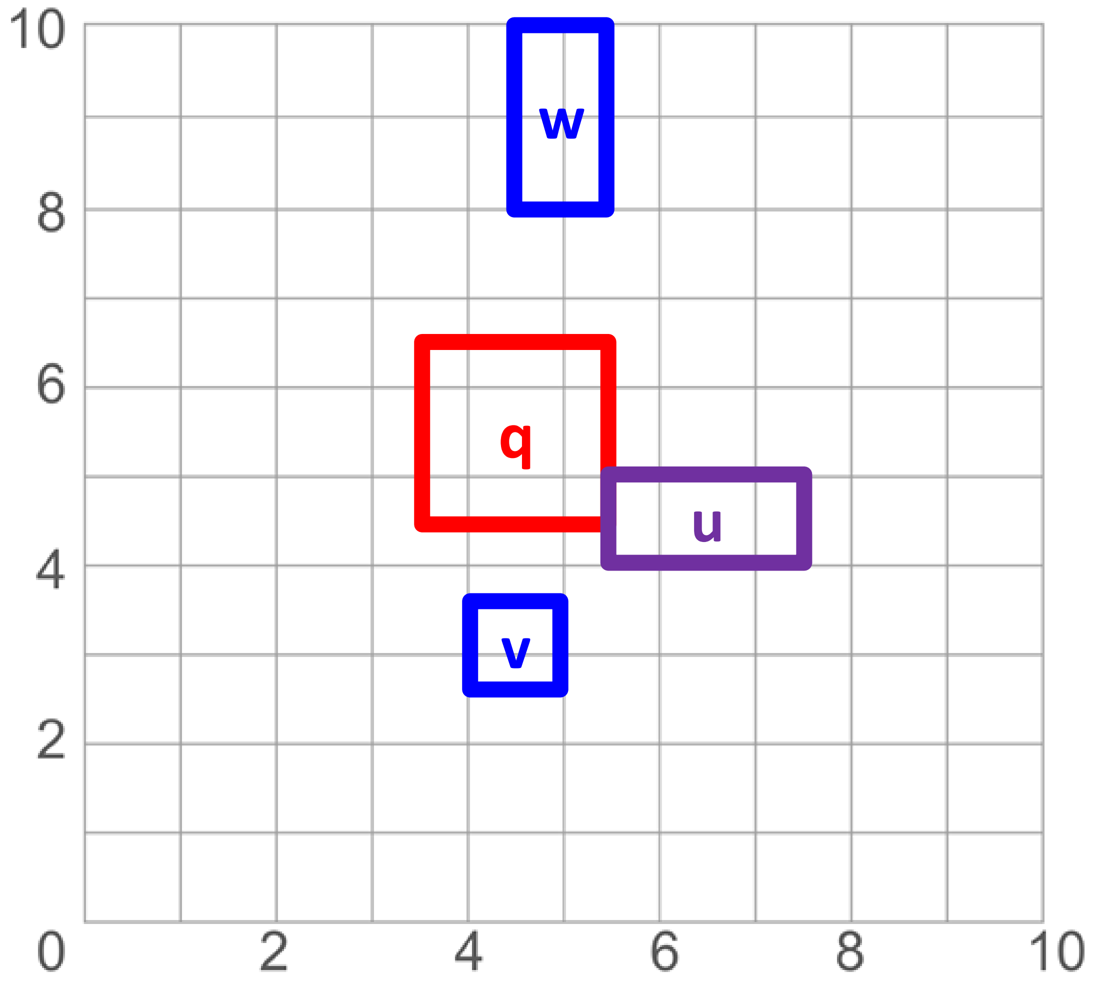
t=0



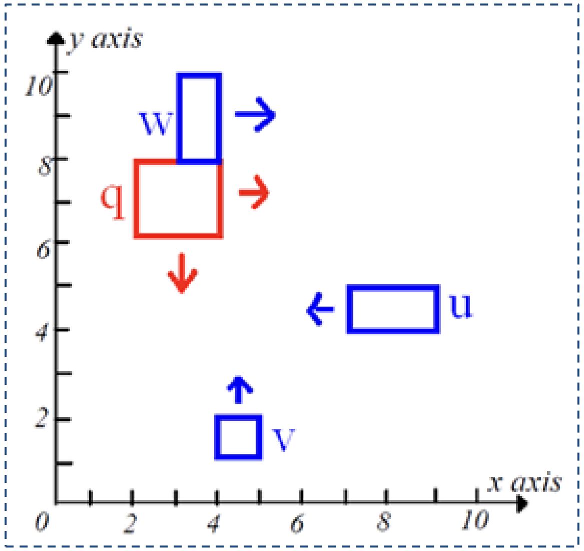


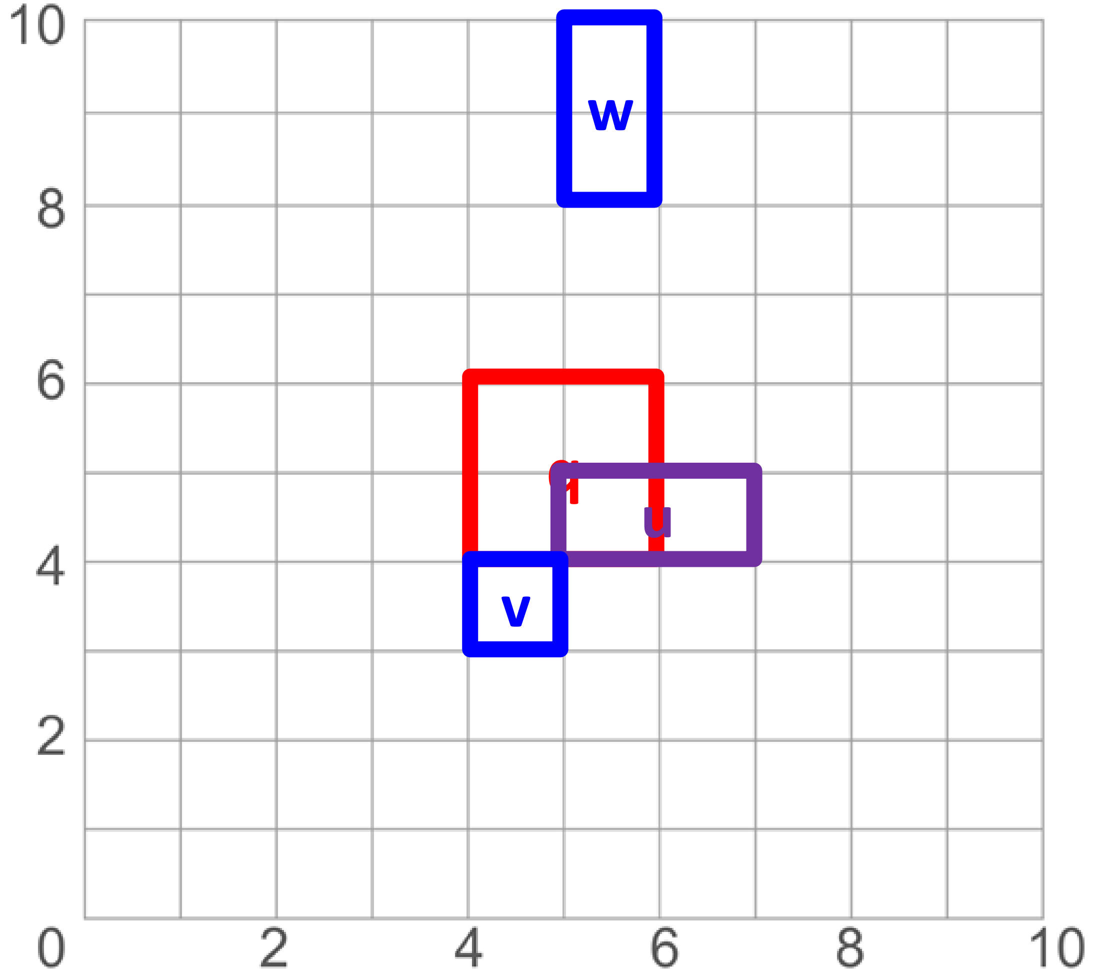
t=1.5



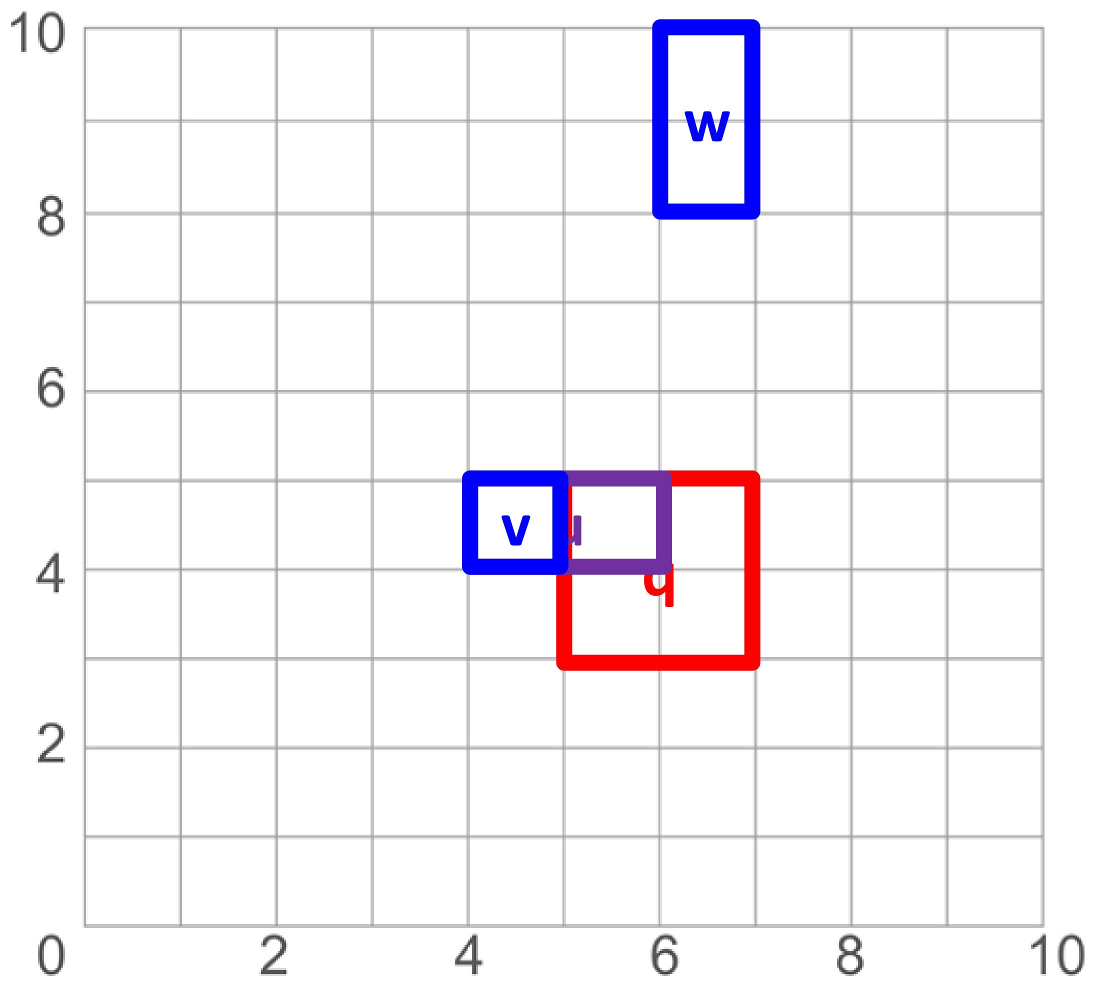


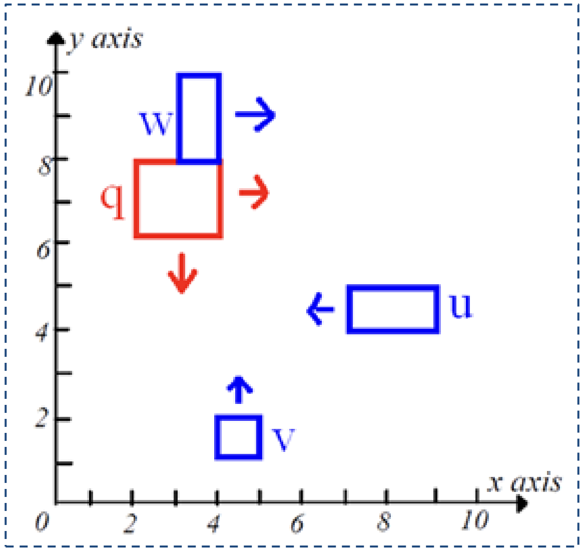
t=2



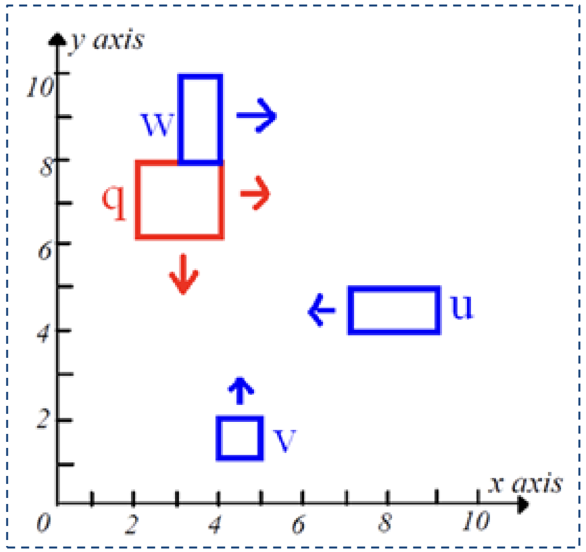


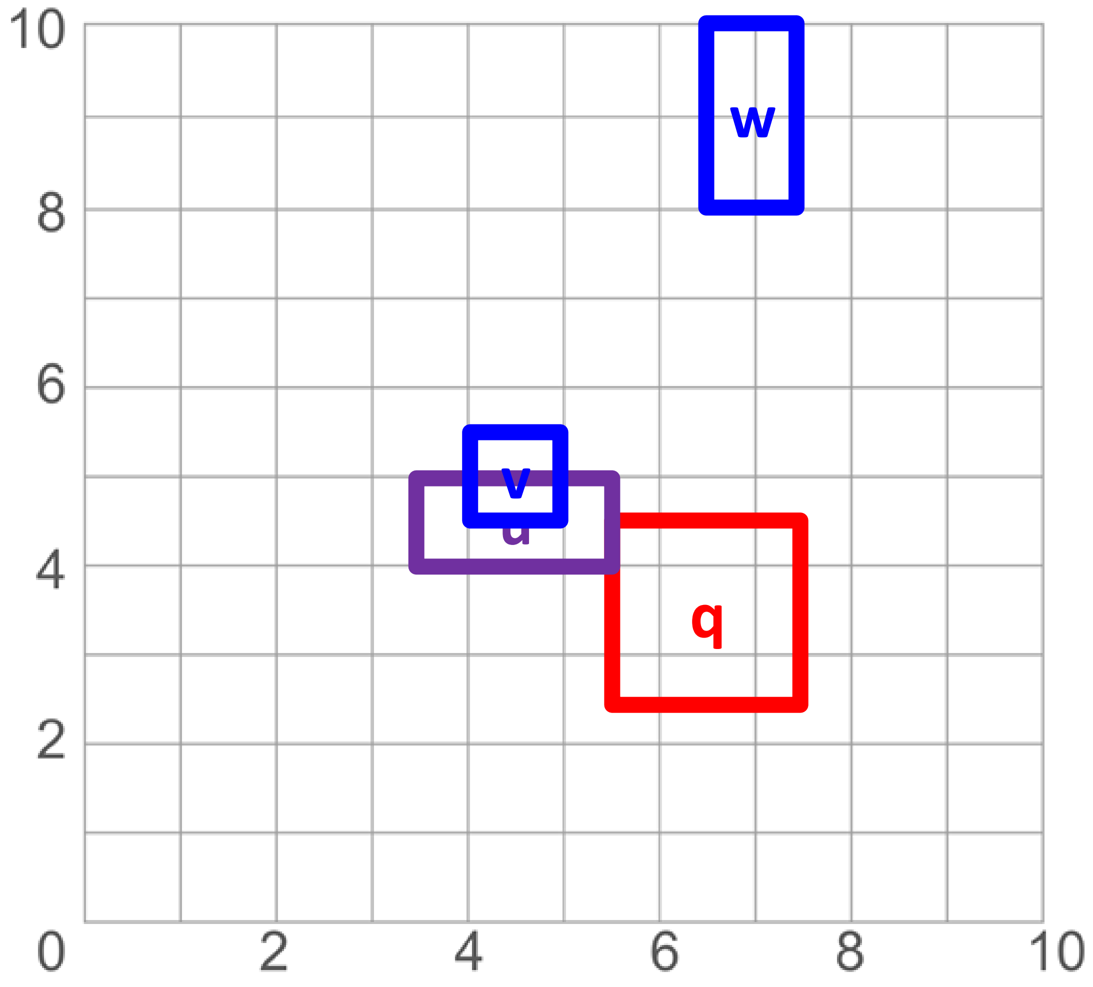
t=3



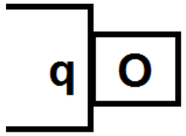
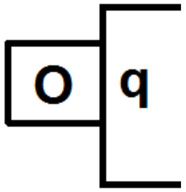


t=3.5





Notation Table

o	An object
q	A query (e.g. window)
$T_{INF}(o, q)$	Influence time := when o will influence q
$[T_s, T_e)$	Intersection period
T_{DSP}	Disappearance time
Subscript i	For dimension i
T_{LR}	Time 
T_{RL}	Time 

TP Queries Output Form <R, T, C>
- R: current result
- T: expiry time
- C: objects that will affect R at time T