Efficient Continuous *K*-Nearest Neighbor Query Processing over Moving Objects with Uncertain Speed and Direction

Yuan-Ko Huang, Shi-Jei Liao, and Chiang Lee

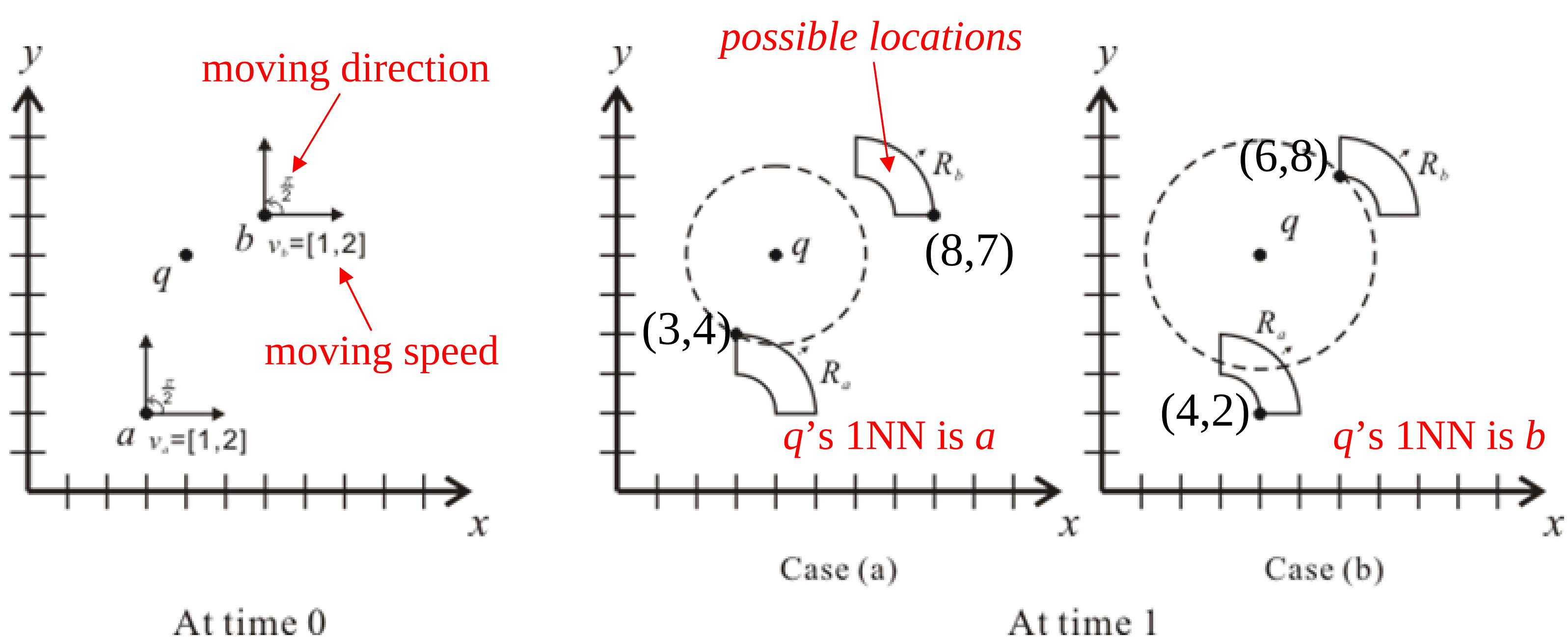
Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, R.O.C.

Introduction (1/3)

- Continuous *K*-Nearest Neighbor (C*K*NN) query:
  - Finding the *K*-nearest neighbors (*K*NNs) of a moving user at each time instant within a user-given time interval $[t_s, t_e]$.
- We focus on how to process such a C*K*NN query on moving objects with uncertain speed and direction.
  - Speed of object is between a minimal and a maximal speed.
  - Direction of object is between a minimal and a maximal angle.

Introduction (2/3)

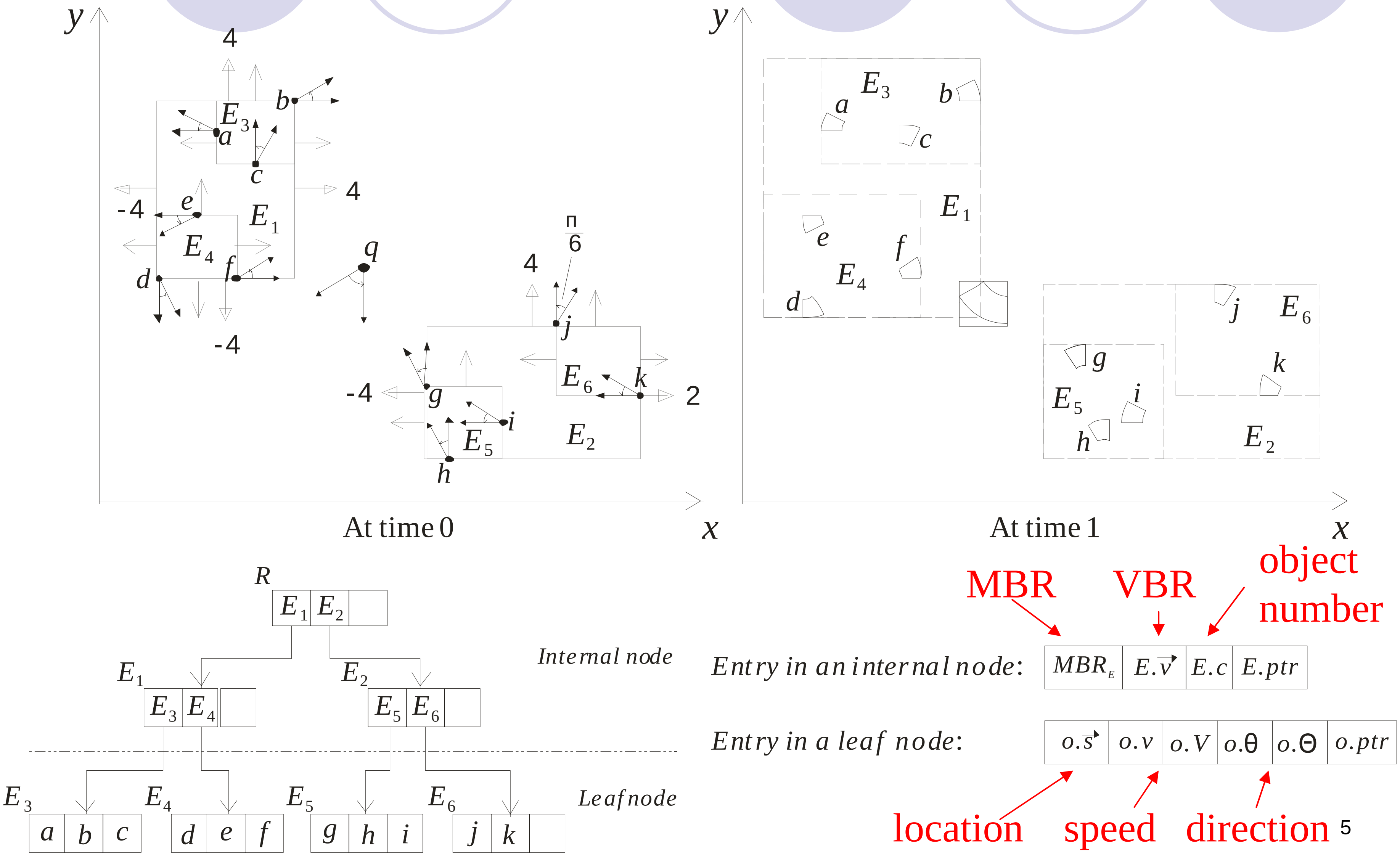
- Effect of uncertain speed and direction on processing C*K*NN query:
  - All "possible" locations should be taken into account so as to guarantee that all *possible KNNs* (P*K*NNs) will be included in the result



Introduction (3/3)

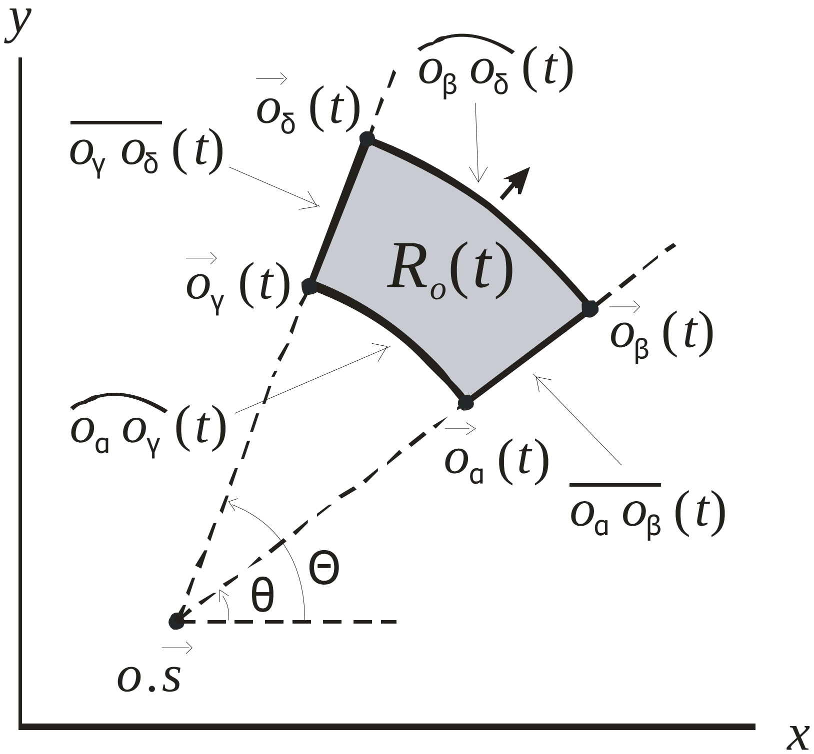
- Contributions:
  - We develop a TPR$^{(s,d)}$-tree to index moving objects with uncertainty.
  - An uncertain distance model is presented for representing the distance between objects.
  - We propose a continuous P*K*NN (CP*K*NN) algorithm to determine the P*K*NNs from t_s to t_e.

TPR$^{(s,d)}$-tree


At time 0
At time 1
Internal node
Leaf node
Entry in an internal node:
Entry in a leaf node:
MBR
VBR
object number
location
speed
direction

Uncertain distance model (1/5)

- Possible region $R_o(t)$ of object o:
  - object o's speed varies within [$o.v$, $o.V$].
  - object o's direction varies within [$o.\theta$, $o.\Theta$]

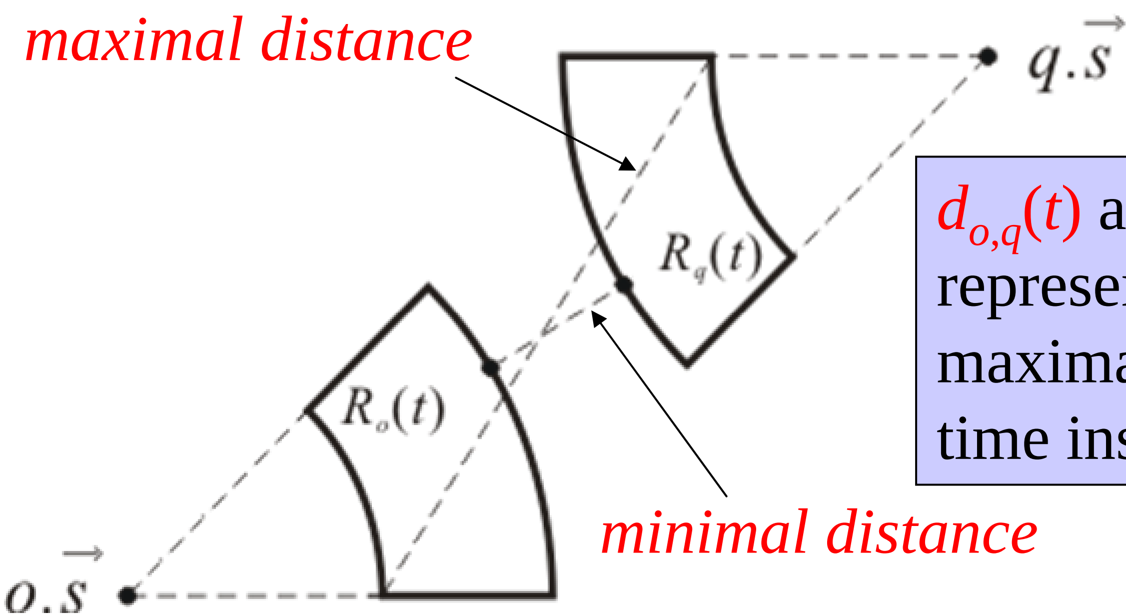


possible region $R_o(t)$ is enclosed by four endpoints, two segments, and two arcs

possible region $R_o(t)$ moves with time

Uncertain distance model (2/5)

- Given $R_o(t)$ and $R_q(t)$, the distance between *o* and *q* at any time *t* will be bounded by a minimal and a maximal distance.

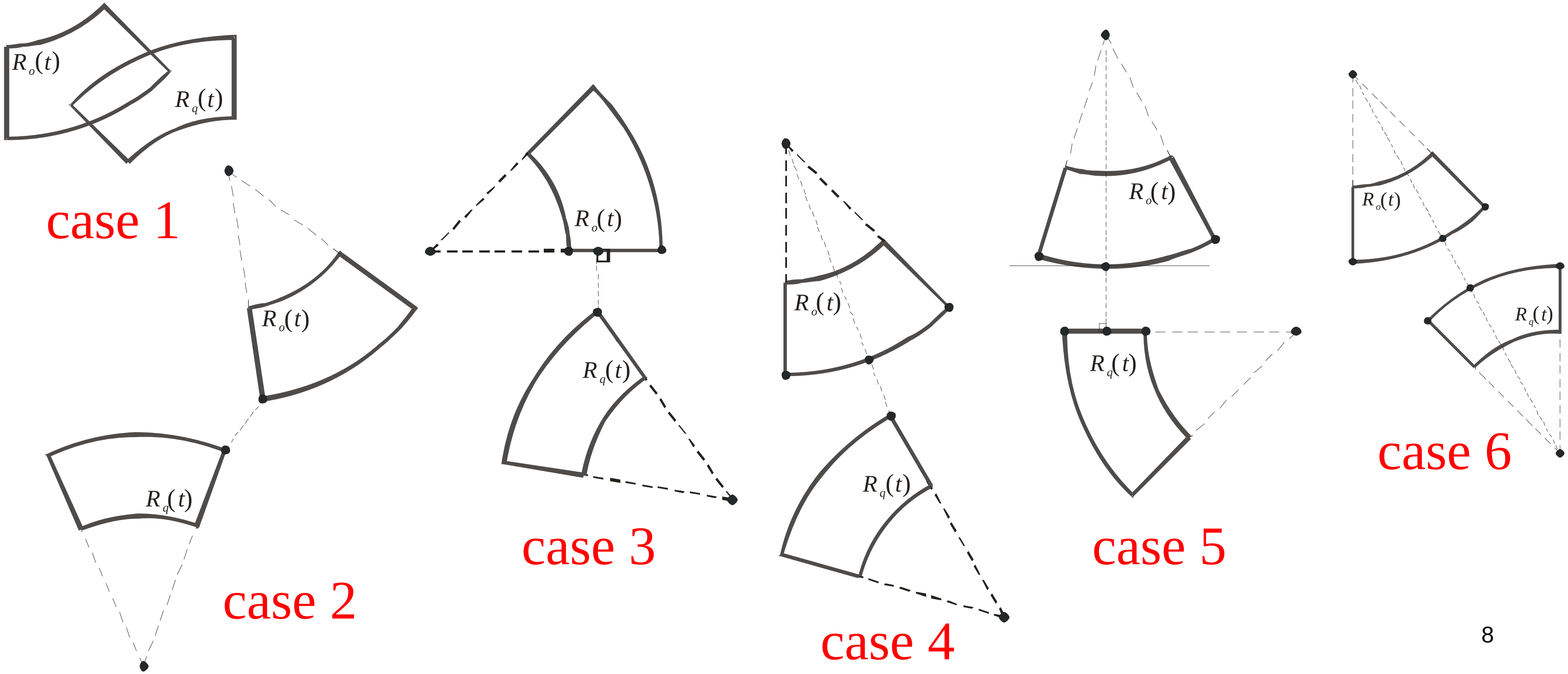


$d_{o,q}(t)$ and $D_{o,q}(t)$ are used to represent the minimal and the maximal distances at each time instant

At time *t*

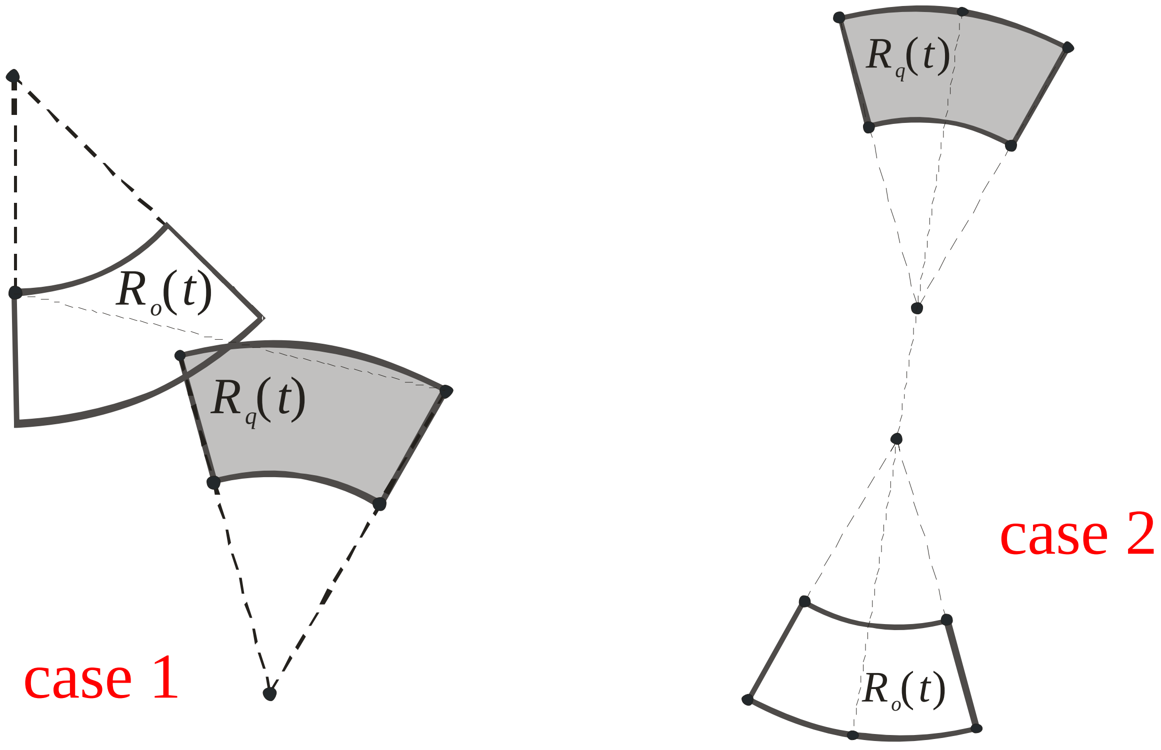
Uncertain distance model (3/5)

- Minimal distance function $d_{o,q}(t)$:
  - at any time instant the minimal distance would belong to one of the six cases



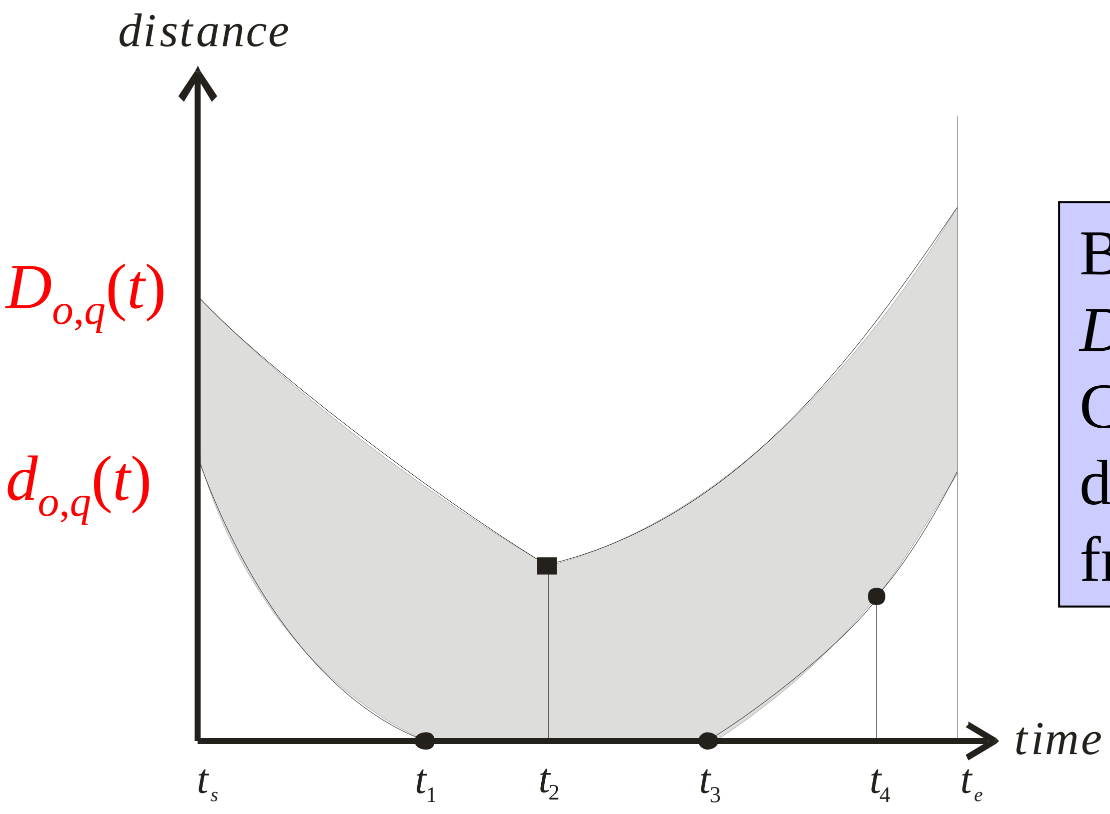
Uncertain distance model (4/5)

- Maximal distance function $D_{o,q}(t)$:
  - at any time instant the maximal distance would belong to one of the two cases



Uncertain distance model (5/5)

- Each point in the region bounded by $d_{o,q}(t)$ and $D_{o,q}(t)$ is a possible distance between o and q

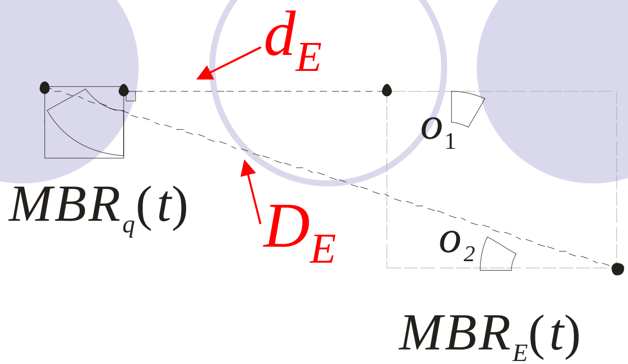


Based on $d_{o,q}(t)$ and $D_{o,q}(t)$, we develop the CP*K*NN algorithm to determine the P*K*NNs from t_s to t_e

CP*K*NN algorithm (1/6)

- Filtering step:
  - employs a branch-and-bound traversal on the TPR$^{(s,d)}$-tree to prune non-qualifying objects
- Refinement step:
  - examines whether the candidates are the P*K*NNs or not

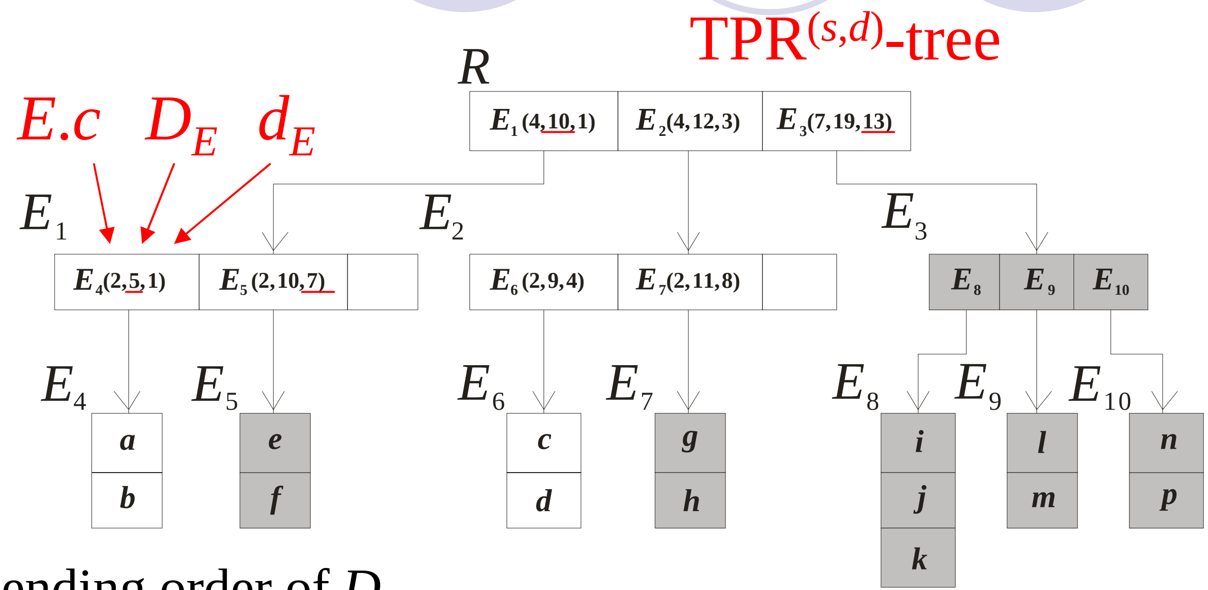
CP*K*NN algorithm (2/6)



- Filtering step:
  - utilizes two parameters to determine whether an index node E is visited or not
    - d_E: the global minimal distance between $MBR_E(t)$ and $MBR_q(t)$ within $[t_s, t_e]$
    - D_E: the global maximal distance between $MBR_E(t)$ and $MBR_q(t)$ within $[t_s, t_e]$
  - pruning heuristic:
    - If there exist n MBRs whose D_E are less than d_{E_i} of $MBR_{E_i}(t)$ and the total number of objects enclosed by these n MBRs is greater than or equal to K, then all of child nodes of E_i can be pruned

CP*K*NN algorithm (3/6)

- Filtering step:
  - Example:
    - finding 2NNs



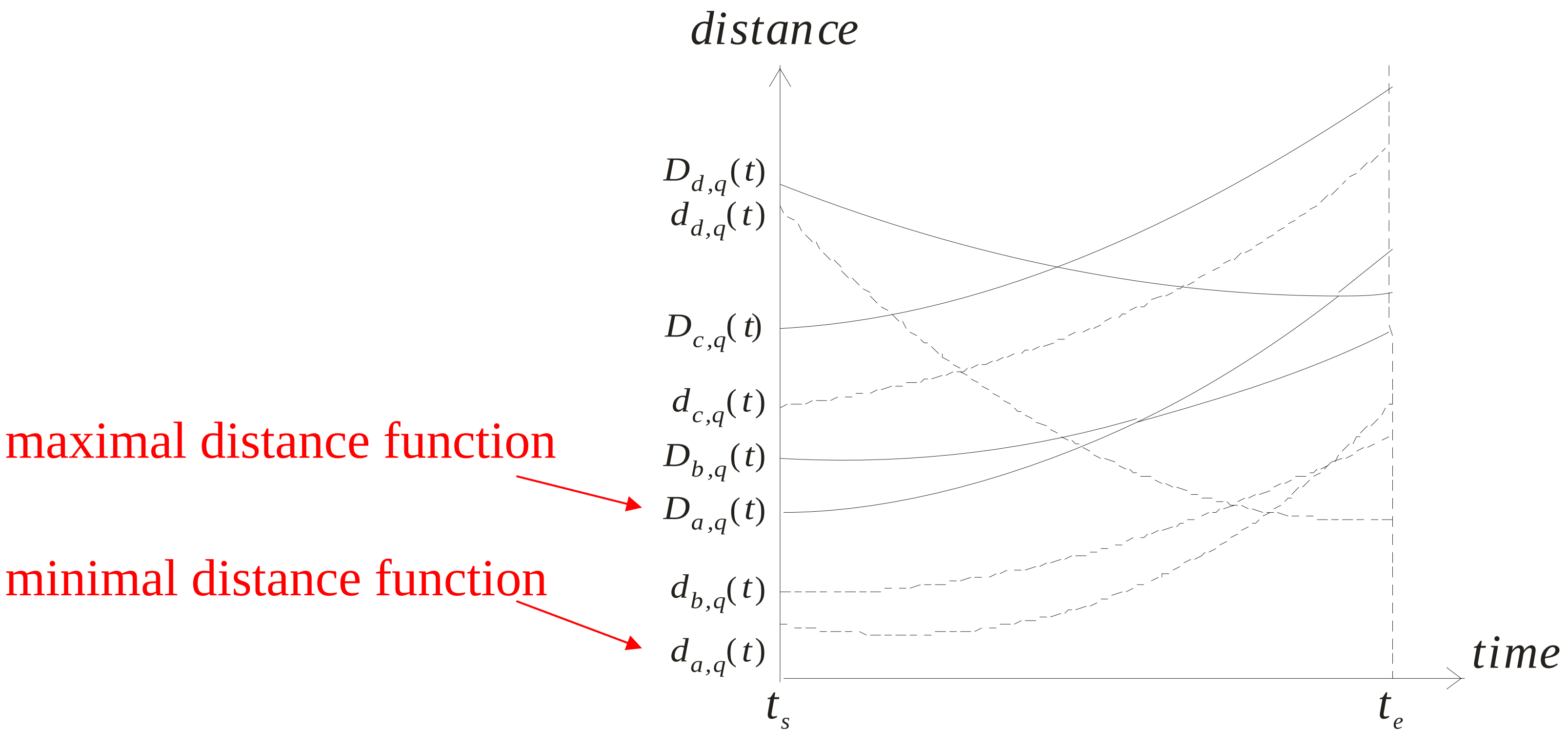
a linked list sorted in ascending order of D_E

Visit R	$L = \{E_1,E_2,E_3\}$	D_{E1}=10	prune E_3
Visit E_1	$L = \{E_4,E_5,E_2\}$	D_{E4}=5	prune E_5
Visit E_4	$L = \{E_2\}$	D_{E4}=5	candidates={a,b}
Visit E_2	$L = \{E_6,E_7\}$	D_{E4}=5	prune E_7
Visit E_6	L = {null}	D_{E4}=5	candidates={a,b,c,d}

objects a, b, c, d will be verified in the refinement step

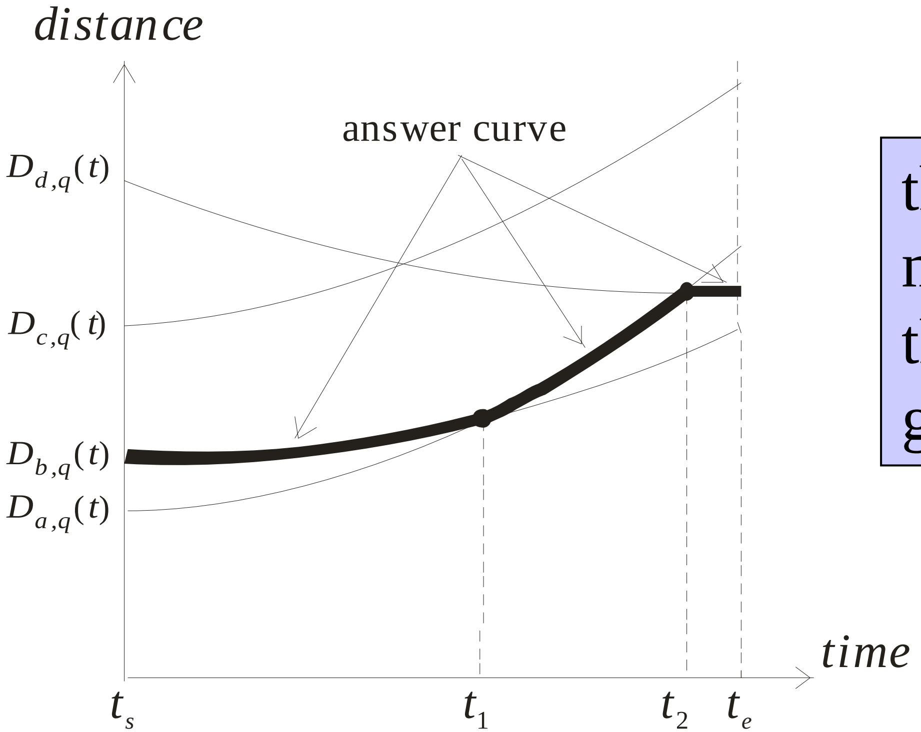
CP*K*NN algorithm (4/6)

- Refinement step:
  - The minimal and maximal distance functions of the candidates will be computed and represented in the time-distance space



CP*K*NN algorithm (5/6)

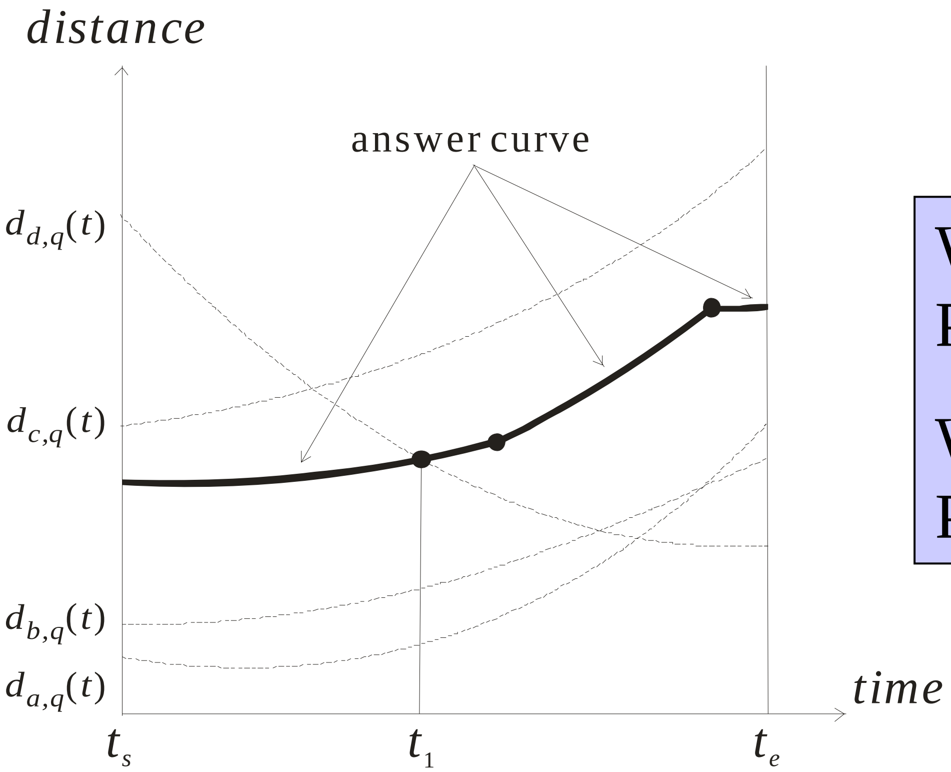
- Refinement step:
  - Heuristic:
    - at each time instant, if there exist *K* candidates whose maximal distances are less than the minimal distance of candidate *o*', then *o*' must not be a P*K*NN



the candidate o_k whose maximal distance ranks at the *K*-th smallest is used to generate an answer-curve

CP*K*NN algorithm (6/6)

- Refinement step:
  - a candidate *o* is a P*K*NN only if its $d_{o,q}(t)$ is below the answer-curve



Within $[t_s, t_1]$, the P*K*NNs are *a* and *b*

Within $[t_1, t_e]$, the P*K*NNs are *a*, *b*, and *d*

Conclusions

- We focused on processing the C*K*NN query over moving objects with uncertain speed and direction
- We proposed an uncertain distance model to formulate the uncertain distance between objects.
- We developed the CP*K*NN algorithm to process a C*K*NN query

Thanks all.