# An Adaptive Updating Protocol for Reducing Moving Object Database Workload

By Su Chen, Beng Chin Ooi, Zhenjie Zhang

Sari Haj Hussein[1]

[1]Department of Computer Science
Aalborg University

2012-05-16

# What is the Paper about?

- The paper is concerned with Moving Object Databases (MODs)
- These MODs keep track of objects by receiving occasional location updates
- The paper proposes a protocol that reduces location update frequency (via approximation), thereby reducing the workload on MODs

# What is the Paper about?

- The paper is concerned with Moving Object Databases (MODs)
- These MODs keep track of objects by receiving occasional location updates
- The paper proposes a protocol that reduces location update frequency (via approximation), thereby reducing the workload on MODs

# Temporal Bounded Strategy

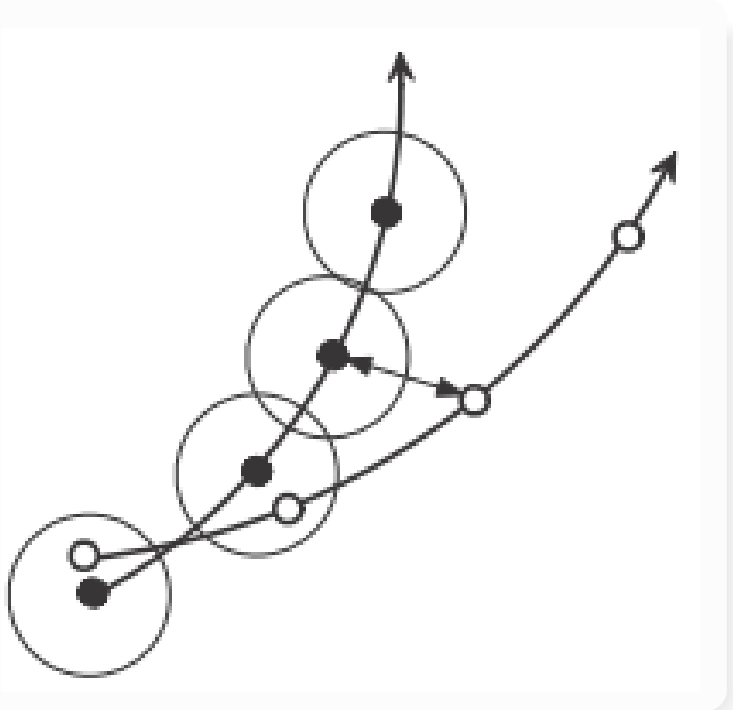
- A moving object updates its motion model with the MOD periodically at fixed time intervals
- Not efficient $\rightsquigarrow$ An update is issued even if the model does not change since the last update

# Temporal Bounded Strategy

- A moving object updates its motion model with the MOD periodically at fixed time intervals
- Not efficient $\rightsquigarrow$ An update is issued even if the model does not change since the last update

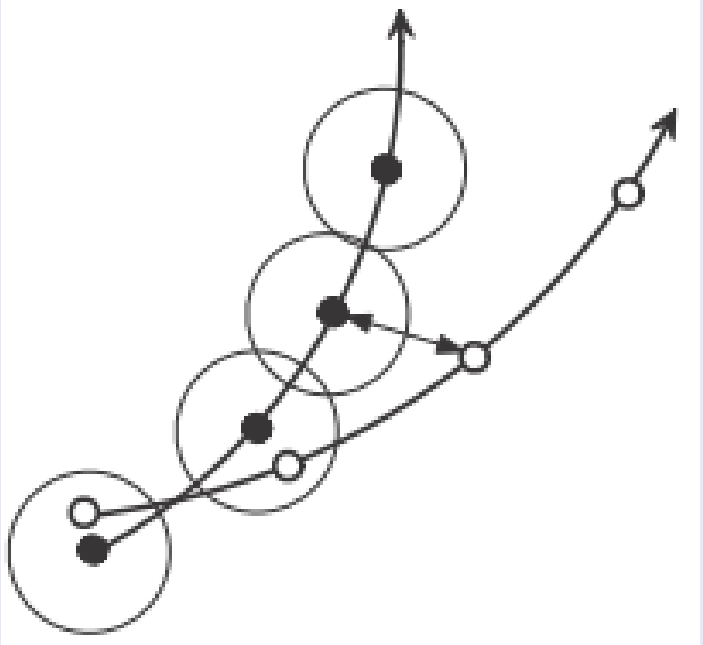
# Spatial Bounded Strategy

- Adaptively decide the update time depending on the spatial error
- Solid points are the predicted locations of the old model (on the MOD-side)
- Hollow points are the predicted locations of the new model (on the object-side)
- Distance between solid and hollow points in the near future exceeds a threshold $\rightsquigarrow$ The object updates its motion model with the MOD



# Spatial Bounded Strategy

- Adaptively decide the update time depending on the spatial error
- Solid points are the predicted locations of the old model (on the MOD-side)
- Hollow points are the predicted locations of the new model (on the object-side)
- Distance between solid and hollow points in the near future exceeds a threshold $\rightsquigarrow$ The object updates its motion model with the MOD



# Assumptions

- A linear motion model of objects
- Location of object $o_i$ at timestamp $t$ is $l_i^t = (l_i^t.x, l_i^t.y)$
- Velocity of object $o_i$ at timestamp $t$ is $v_i^t = (v_i^t.x, v_i^t.y)$
- Predicted location of $o_i$ at $s \geq t$ is
  $pl_i^s = (pl_i^s.x, pl_i^s.y) = (l_i^t.x + v_i^t.x(s-t), l_i^t.y + v_i^t.y(s-t))$

# Assumptions

- A linear motion model of objects
- Location of object $o_i$ at timestamp $t$ is $l_i^t = (l_i^t.x, l_i^t.y)$
- Velocity of object $o_i$ at timestamp $t$ is $v_i^t = (v_i^t.x, v_i^t.y)$
- Predicted location of $o_i$ at $s \geq t$ is $pl_i^s = (pl_i^s.x, pl_i^s.y) = (l_i^t.x + v_i^t.x(s-t), l_i^t.y + v_i^t.y(s-t))$

# Assumptions

- A linear motion model of objects
- Location of object $o_i$ at timestamp $t$ is $l_i^t = (l_i^t.x, l_i^t.y)$
- Velocity of object $o_i$ at timestamp $t$ is $v_i^t = (v_i^t.x, v_i^t.y)$
- Predicted location of $o_i$ at $s \geq t$ is
  $pl_i^s = (pl_i^s.x, pl_i^s.y) = (l_i^t.x + v_i^t.x(s-t), l_i^t.y + v_i^t.y(s-t))$

# Spatio-Temporal Safe Region (STSR)

## Spatio-Temporal Safe Region (STSR)

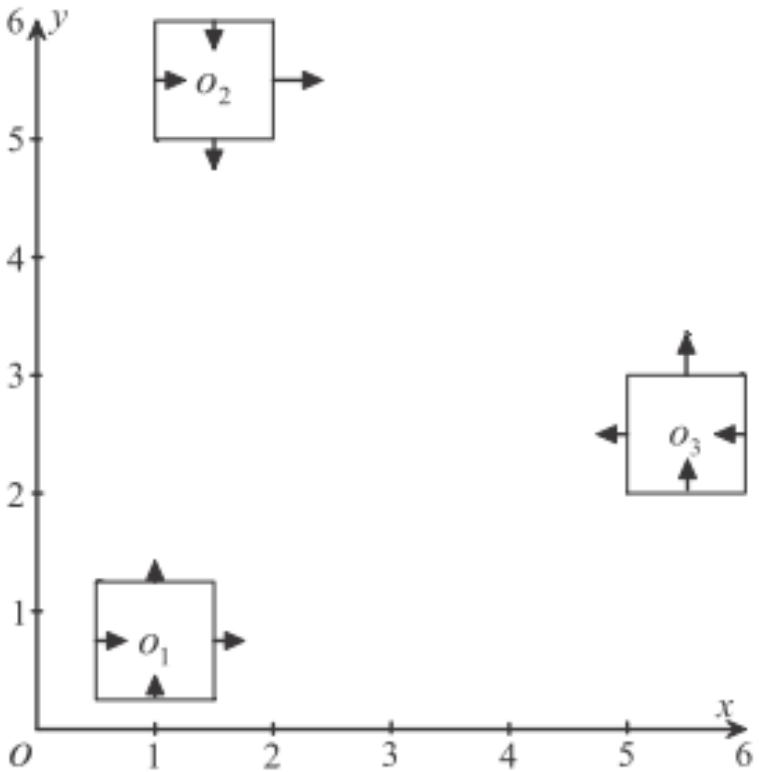
- A rectangle in the spatio-temporal space bounding the location and velocity of a moving object
- One and only one STSR per moving object
- An STSR is stored on the client device as well as on the MOD
- $R(o_i) = (LR, VR, t_r, t_e)$
  - $LR = [LR.x^{\vdash}, LR.x^{\dashv}] \times [LR.y^{\vdash}, LR.y^{\dashv}]$ rectangle in the physical space
  - $VR = [VR.x^{\vdash}, VR.x^{\dashv}] \times [VR.y^{\vdash}, VR.y^{\dashv}]$ rectangle is the velocity space
  - $t_r$ reference time
  - $t_e$ expiry time

# Spatio-Temporal Safe Region (STSR)

## Spatio-Temporal Safe Region (STSR)

- A rectangle in the spatio-temporal space bounding the location and velocity of a moving object
- One and only one STSR per moving object
- An STSR is stored on the client device as well as on the MOD
- $R(o_i) = (LR, VR, t_r, t_e)$
  - $LR = [LR.x^{\vdash}, LR.x^{\dashv}] \times [LR.y^{\vdash}, LR.y^{\dashv}]$ rectangle in the physical space
  - $VR = [VR.x^{\vdash}, VR.x^{\dashv}] \times [VR.y^{\vdash}, VR.y^{\dashv}]$ rectangle is the velocity space
  - $t_r$ reference time
  - $t_e$ expiry time

# Example




| STSR | $LR$ | $VR$ | $t_r$ | $t_e$ |
|---|---|---|---|---|
| $R(o_1)$ | $[0.5, 1.5] \times [0.2, 1.2]$ | $[0.5, 0.5] \times [1, 1]$ | 1 | 4 |
| $R(o_2)$ | $[1, 2] \times [5, 6]$ | $[1.2, 2] \times [-1.4, -1]$ | 2 | 5 |
| $R(o_3)$ | $[5, 6] \times [2, 3]$ | $[-1, -1] \times [0.5, 0.75]$ | 1 | 5 |

# Predicted Region

## Predicted Region

- The maximal spatial rectangle expanded from $LR$ with respect to $VR$

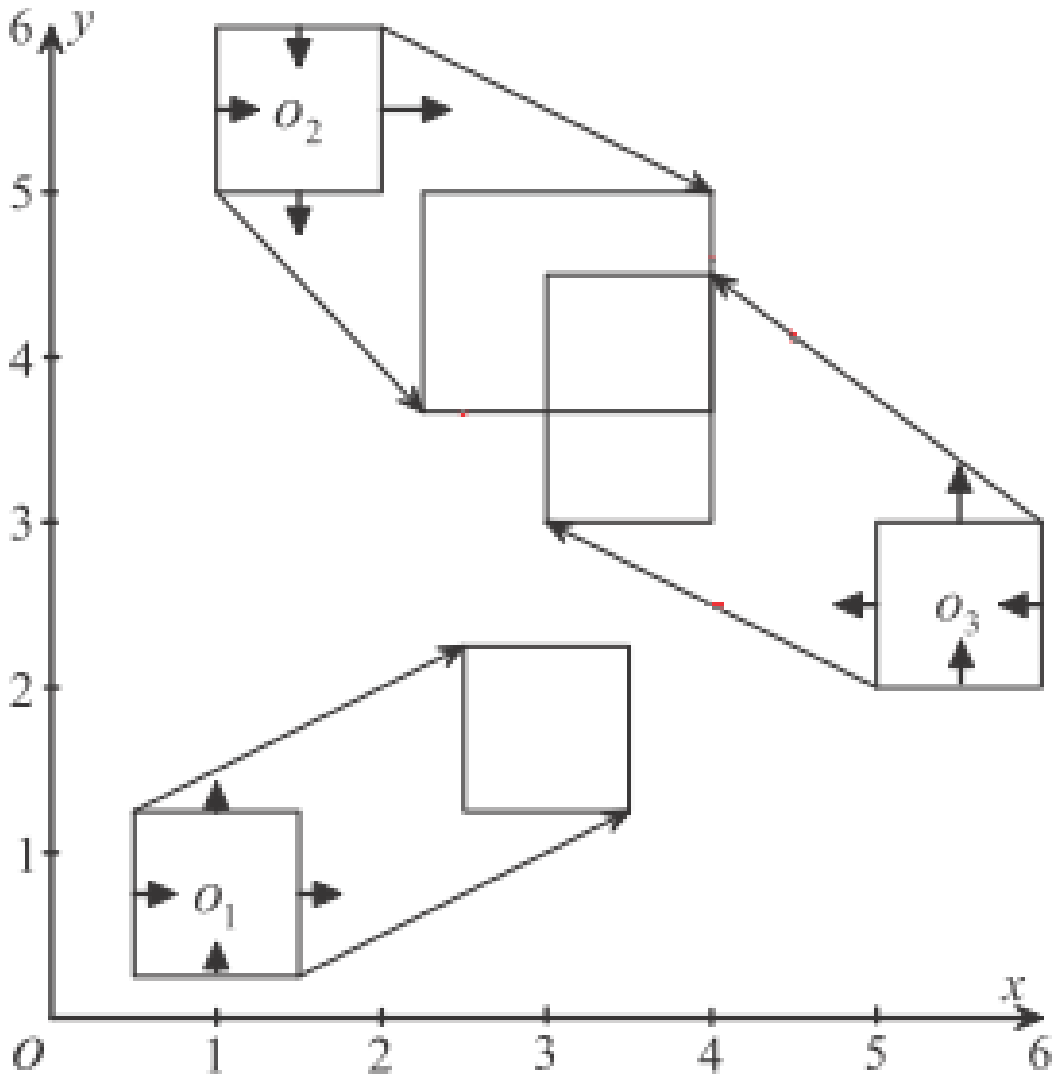
- Given $R(o_i) = (LR, VR, t_r, t_e)$ and $t_r \leq t \leq t_e$
- $P_i^t = [P.x^{\vdash}, P.x^{\dashv}] \times [P.y^{\vdash}, P.y^{\dashv}]$
  - $P.x^{\vdash} = LR.x^{\vdash} + VR.x^{\vdash}(t - t_r)$
  - $P.x^{\dashv} = LR.x^{\dashv} + VR.x^{\dashv}(t - t_r)$
  - $P.y^{\vdash} = LR.y^{\vdash} + VR.y^{\vdash}(t - t_r)$
  - $P.y^{\dashv} = LR.y^{\dashv} + VR.y^{\dashv}(t - t_r)$

# Predicted Region

## Predicted Region

- The maximal spatial rectangle expanded from $LR$ with respect to $VR$
- Given $R(o_i) = (LR, VR, t_r, t_e)$ and $t_r \leq t \leq t_e$
- $P_i^t = [P.x^{\vdash}, P.x^{\dashv}] \times [P.y^{\vdash}, P.y^{\dashv}]$
  - $P.x^{\vdash} = LR.x^{\vdash} + VR.x^{\vdash}(t - t_r)$
  - $P.x^{\dashv} = LR.x^{\dashv} + VR.x^{\dashv}(t - t_r)$
  - $P.y^{\vdash} = LR.y^{\vdash} + VR.y^{\vdash}(t - t_r)$
  - $P.y^{\dashv} = LR.y^{\dashv} + VR.y^{\dashv}(t - t_r)$

# Example



# STSR Consistency

## STSR Consistency

An STSR $R(o_i)$ is consistent with $o_i$ at $t \leq t_e$ if both:

1. Current location $l_i^t$ remains within $P_i^t$ derived from $R(o_i)$
2. For any $t < s \leq t_e$, the predicted location $pl_i^s$ remains within $P_i^s$ derived from $R(o_i)$

## Notes

- Consistency only depends on location - not on velocity
- $R(o_i)$ is consistent with $o_i$ at $t \leq t_e \Leftrightarrow o_i$ is "safe" at $t \leq t_e$
- A consistency verification algorithm for that appears in the paper (true/false)

# STSR Consistency

## STSR Consistency

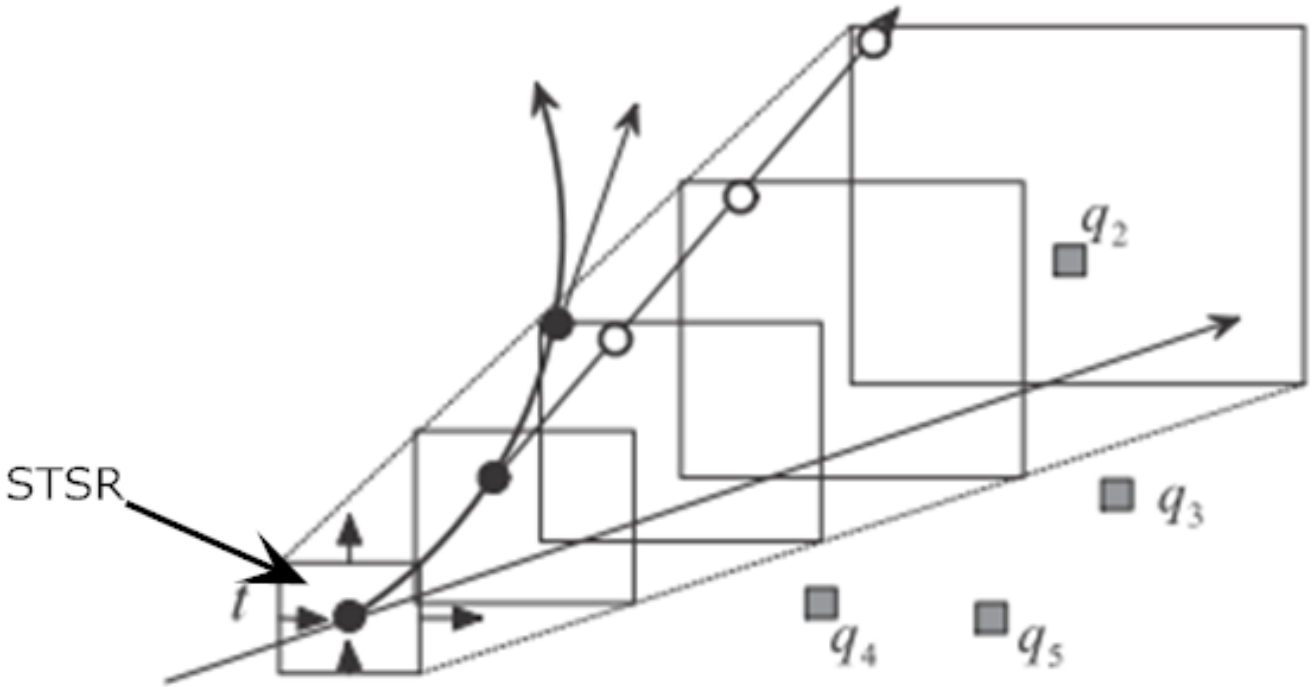
An STSR $R(o_i)$ is consistent with $o_i$ at $t \leq t_e$ if both:

1. Current location $l_i^t$ remains within $P_i^t$ derived from $R(o_i)$
2. For any $t < s \leq t_e$, the predicted location $pl_i^s$ remains within $P_i^s$ derived from $R(o_i)$

## Notes

- Consistency only depends on location - not on velocity
- $R(o_i)$ is consistent with $o_i$ at $t \leq t_e \Leftrightarrow o_i$ is "safe" at $t \leq t_e$
- A consistency verification algorithm for that appears in the paper (true/false)

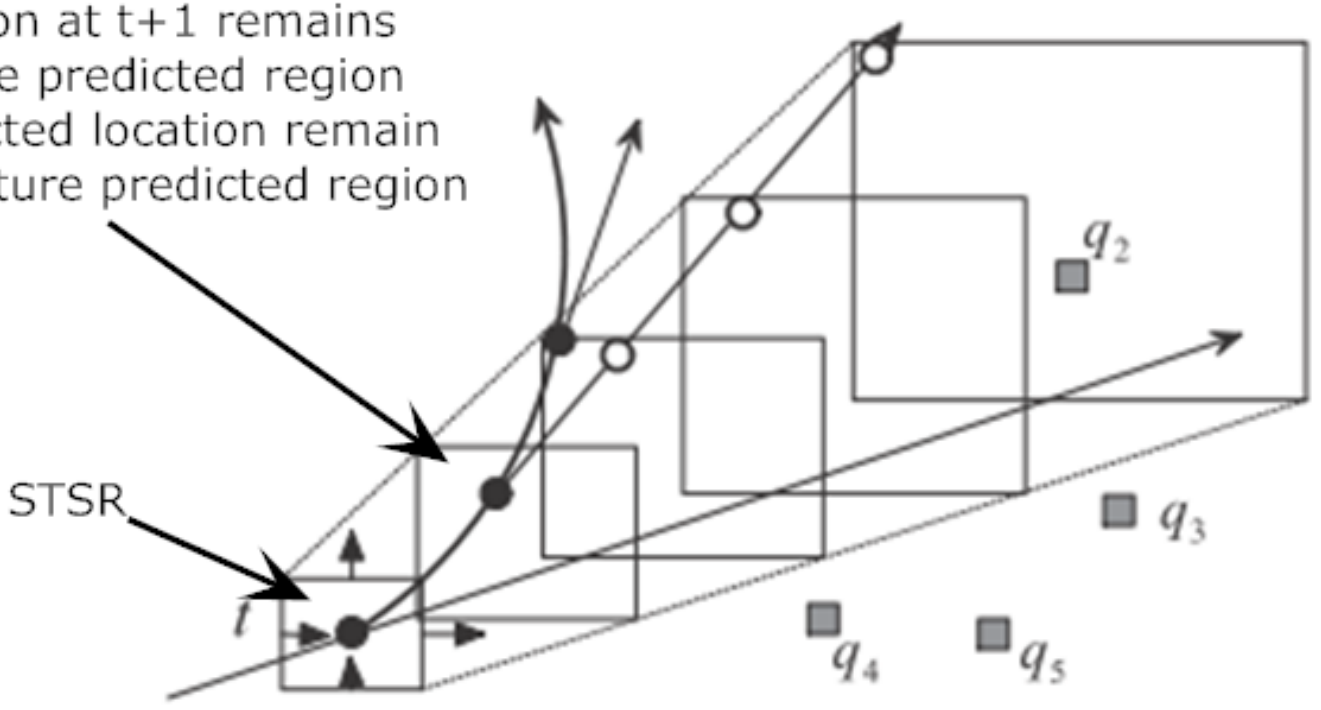
# High-level Illustration



# High-level Illustration

Object is safe at t+1 since

- its location at t+1 remains within the predicted region
- its predicted location remain within future predicted region



# Active Updates

## Active Updates

- Initiated by the moving objects themselves
- On each timestamp, the object checks the consistency of the previous STSR using the verification algorithm
- The previous STSR is no longer consistent with the current object's location and velocity $\Rightarrow$ The object issues an active update to the MOD consisting of its current location and velocity
- The MOD creates a new STSR
- The MOD sends the new STSR to the object
- The object's record is updated in the MOD

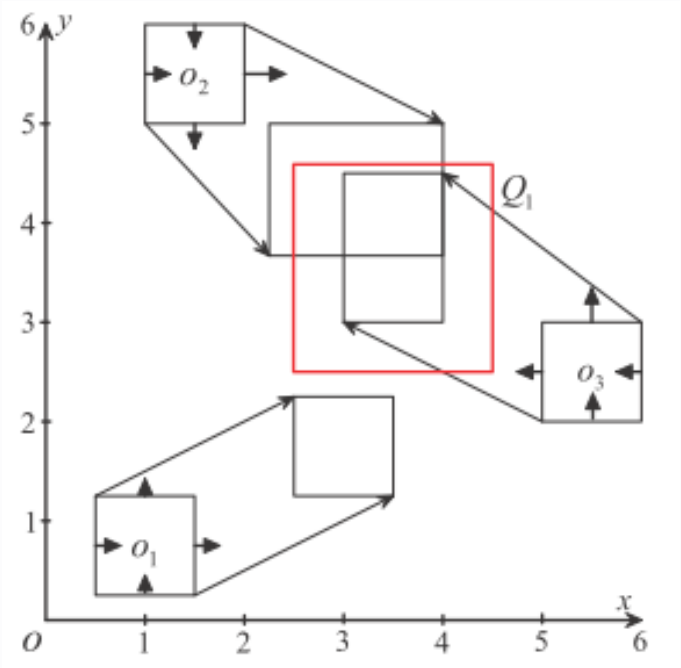
# Active Updates

## Active Updates

- Initiated by the moving objects themselves
- On each timestamp, the object checks the consistency of the previous STSR using the verification algorithm
- The previous STSR is no longer consistent with the current object's location and velocity $\Rightarrow$ The object issues an active update to the MOD consisting of its current location and velocity
- The MOD creates a new STSR
- The MOD sends the new STSR to the object
- The object's record is updated in the MOD

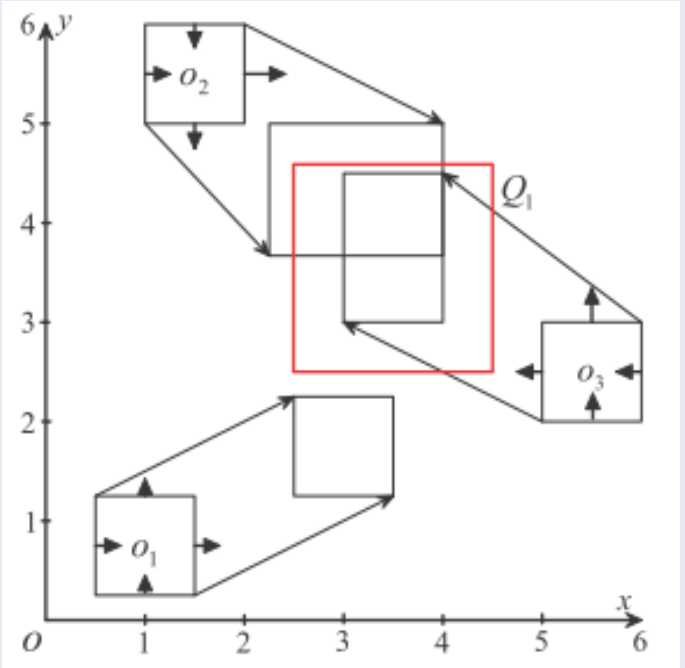
# Predictive Range Query

- Given a querying rectangle $QR$ in the location space and a query time $t_q$, decide all the objects with predicted locations in $QR$ at $t_q$



# Predictive Range Query

- Given a querying rectangle $QR$ in the location space and a query time $t_q$, decide all the objects with predicted locations in $QR$ at $t_q$

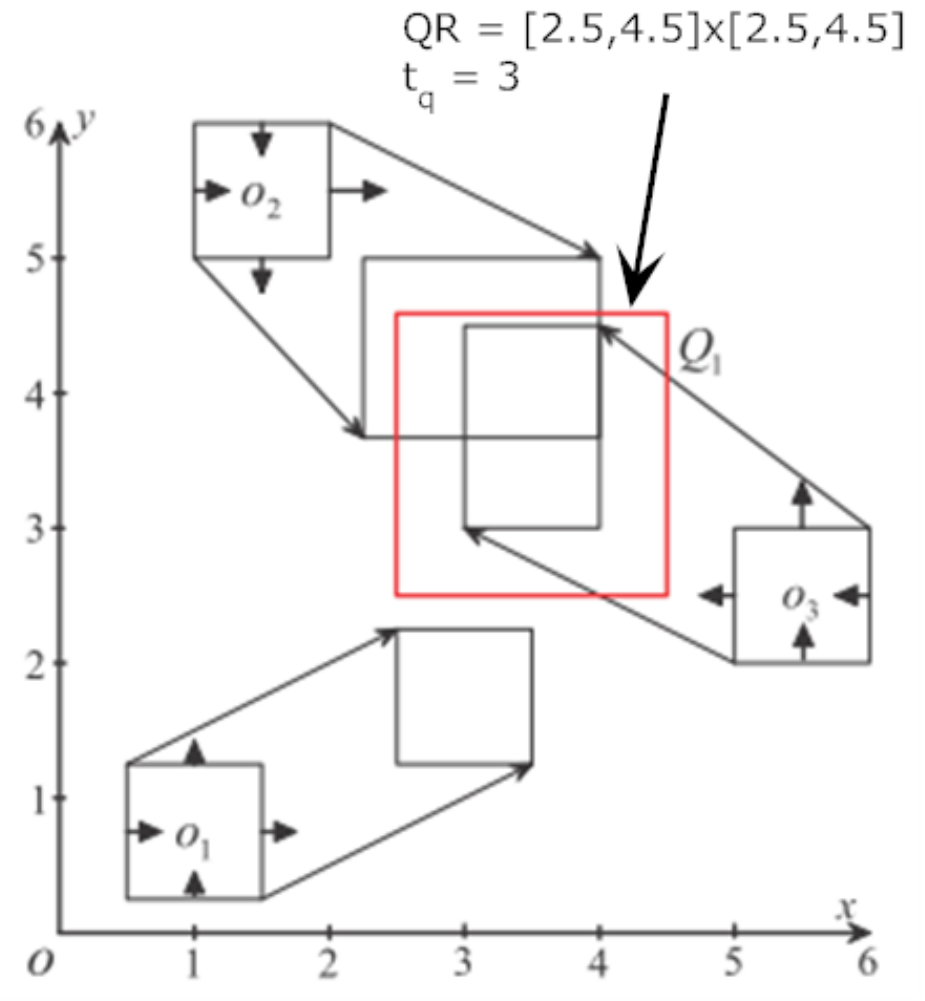


# Passive Updates and Query Processing

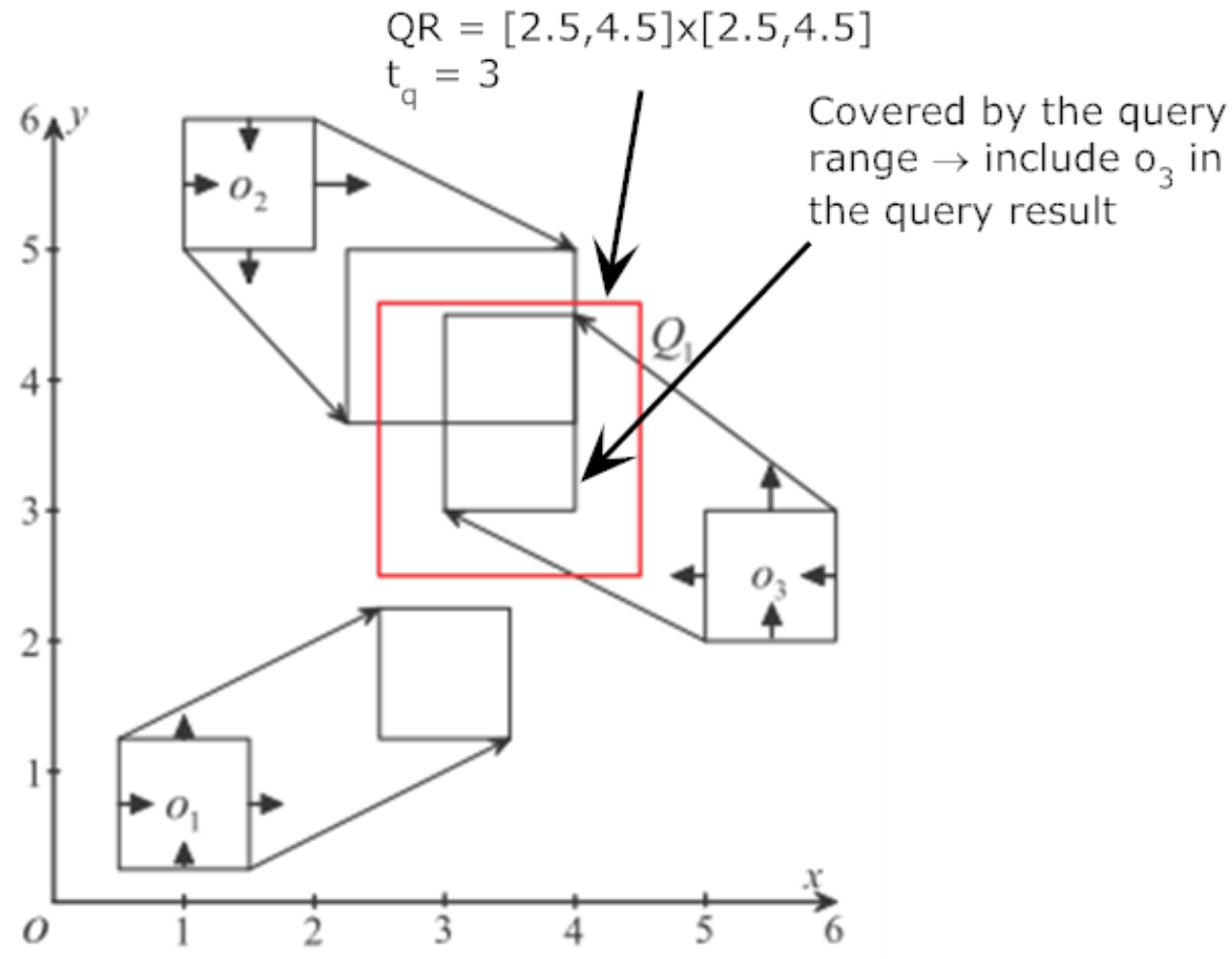
## Passive Updates

- Issued when the MOD processes a predictive range query
- The predicted region partially overlaps with the query region
  $\Rightarrow$ The MOD issues a passive update to the object asking for its current location and velocity
- The MOD uses that to achieve a more accurate prediction in answering the predictive query
- The MOD creates a new STSR
- A query processing algorithm for that appears in the paper (include/exclude objects in/from the query result)

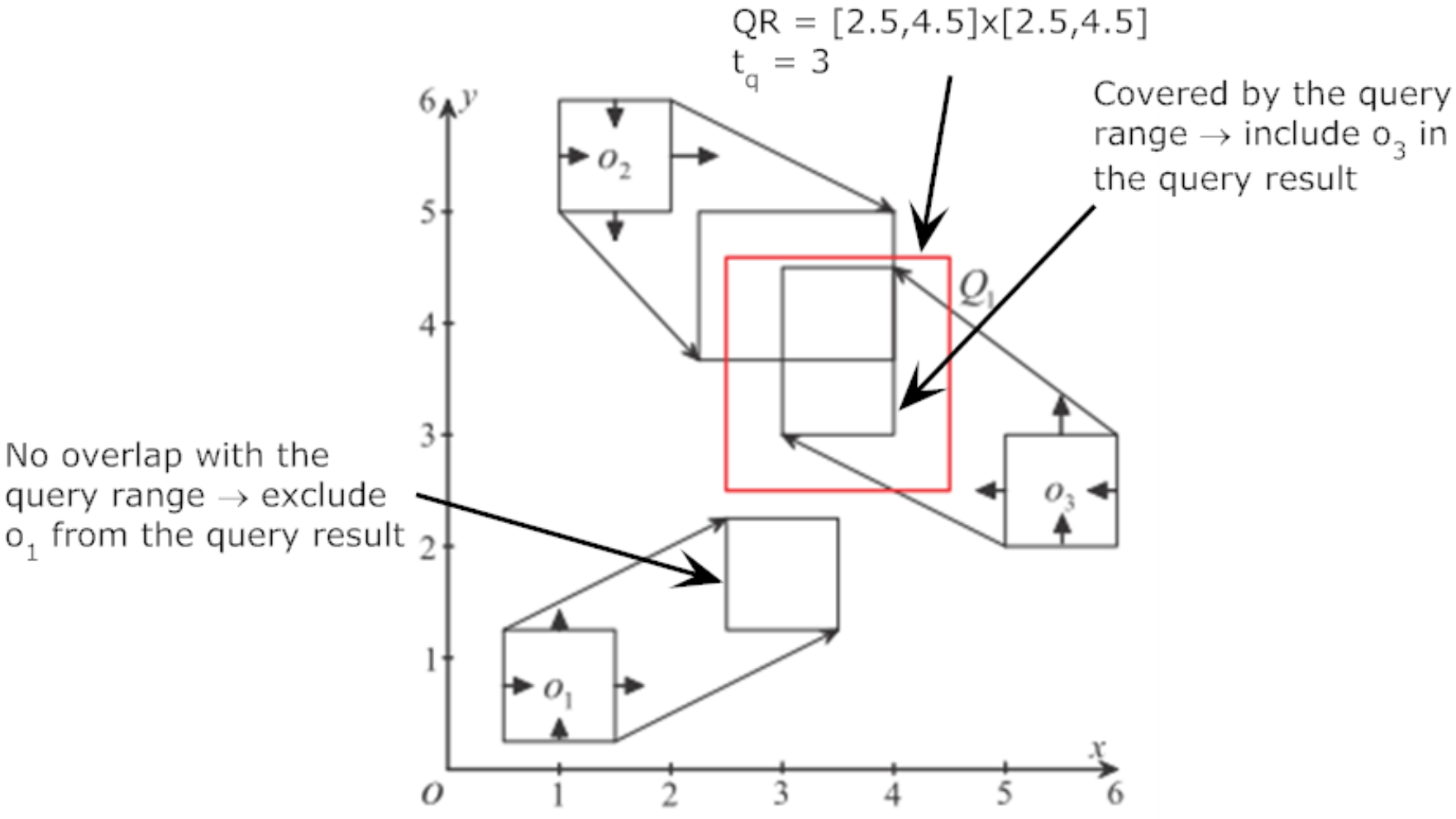
# Example



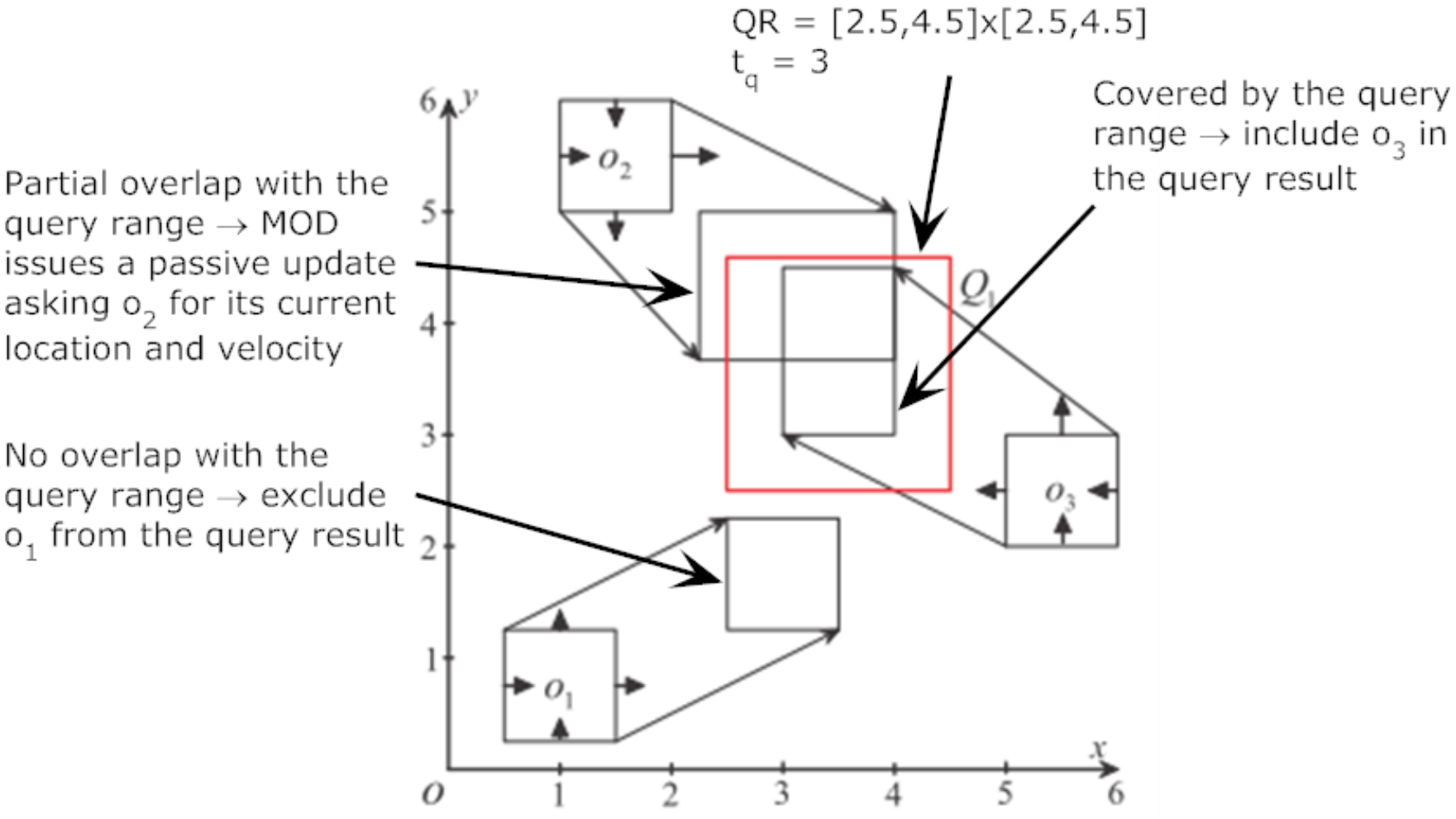
# Example


QR = [2.5,4.5]x[2.5,4.5]
$t_q$ = 3
Covered by the query range → include $o_3$ in the query result
$Q_1$
$o_1$
$o_2$
$o_3$
y
x
O
1
2
3
4
5
6

# Example


QR = [2.5,4.5]x[2.5,4.5]
$t_q$ = 3
Covered by the query range → include $o_3$ in the query result
No overlap with the query range → exclude $o_1$ from the query result
$o_1$
$o_2$
$o_3$
$Q_1$
y
x
O
1
2
3
4
5
6

# Example


QR = [2.5,4.5]x[2.5,4.5]
$t_q$ = 3
Covered by the query range → include $o_3$ in the query result
Partial overlap with the query range → MOD issues a passive update asking $o_2$ for its current location and velocity
No overlap with the query range → exclude $o_1$ from the query result
$o_1$
$o_2$
$o_3$
$Q_1$
y
x
O
1
2
3
4
5
6

# Thank You!