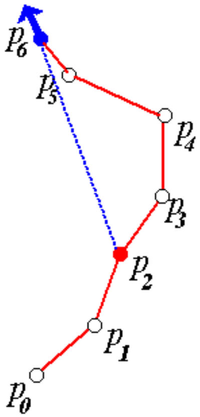
Multi-scale Windowing over Trajectory Streams

Kostas Patroumpas

School of Electrical and Computer Engineering
National Technical University of Athens, Hellas

GeoStreaming

- *Geospatial streams* derived from real-time data acquisition
  - geosensors ~ *vector data* • imagery/satellite ~ *raster data* (mostly)
- Interest on monitoring *location-aware* moving objects:
  – *numerous* vehicles, people, merchandise, animals,...
  - *PRESENT* → record their current *location*
  - *PAST* → maintain historical *trajectories*
  - *FUTURE* → predict routes / estimate trends



- Streaming locations from GPS/RFID/GSM…
  – *timestamped, georeferenced points* pose challenges:
    - ➢ consume fluctuating, intermittent, voluminous positional updates
    - ➢ provide timely response to spatiotemporal *Continuous Queries (CQ)*
    - ➢ overcome lack of suitable operators in traditional DBMS

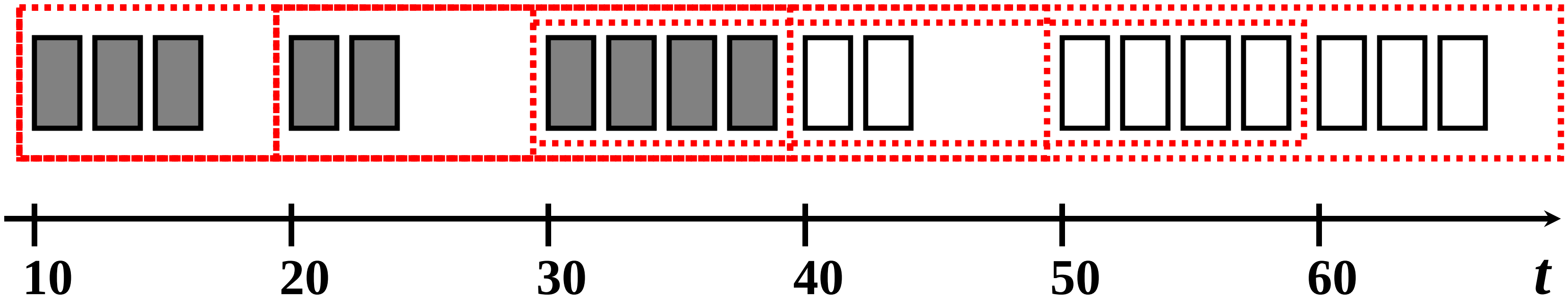
➢ Towards a new processing paradigm: *GeoStreaming*

– query evaluation – in-memory indexing – data reduction/approximation

Windowing over Data Streams

- Continuous queries evaluated against stream "chunks"
  - Infeasible to retain the entire stream "history"
    - for online computation in *main memory*, not disk-bound storage
  - *Windows* provide a *temporary, bounded portion* of a stream:
    - Sliding
    - Tumbling
    - Partitioned
    - Landmark …
  - Typically *sliding windows* inherent in continuous queries:
    - CQ: "*every 10 minutes get average temperature over past 30 minutes*"

```
SELECT AVG(S.temperature)
FROM S [RANGE 30 MINUTES
        SLIDE 10 MINUTES]
```



An "*Amnesic*" Approach

- Handling trajectories of moving objects *online*
  - *Do we need all details of each historical trace?*
    - Significance of each isolated position is *time-decaying*
    - Recent positions are far more important
    - Older segments may be compressed or even purged
- Introduce multi-scale windowing over *trajectory streams*
  - Focus on motion over *varying time horizons* in the past
    - Gradually coarser representations over greater time periods
    - Higher precision reserved for most recent segments
- Core idea extends *multi-granular* windows [TIME'10]
  - Novel "*scale*" semantics for trajectory simplification
    - Obtain generalized, comparable traces for querying
    - Convey reliable motion characteristics
    - No matter the reporting frequency from objects

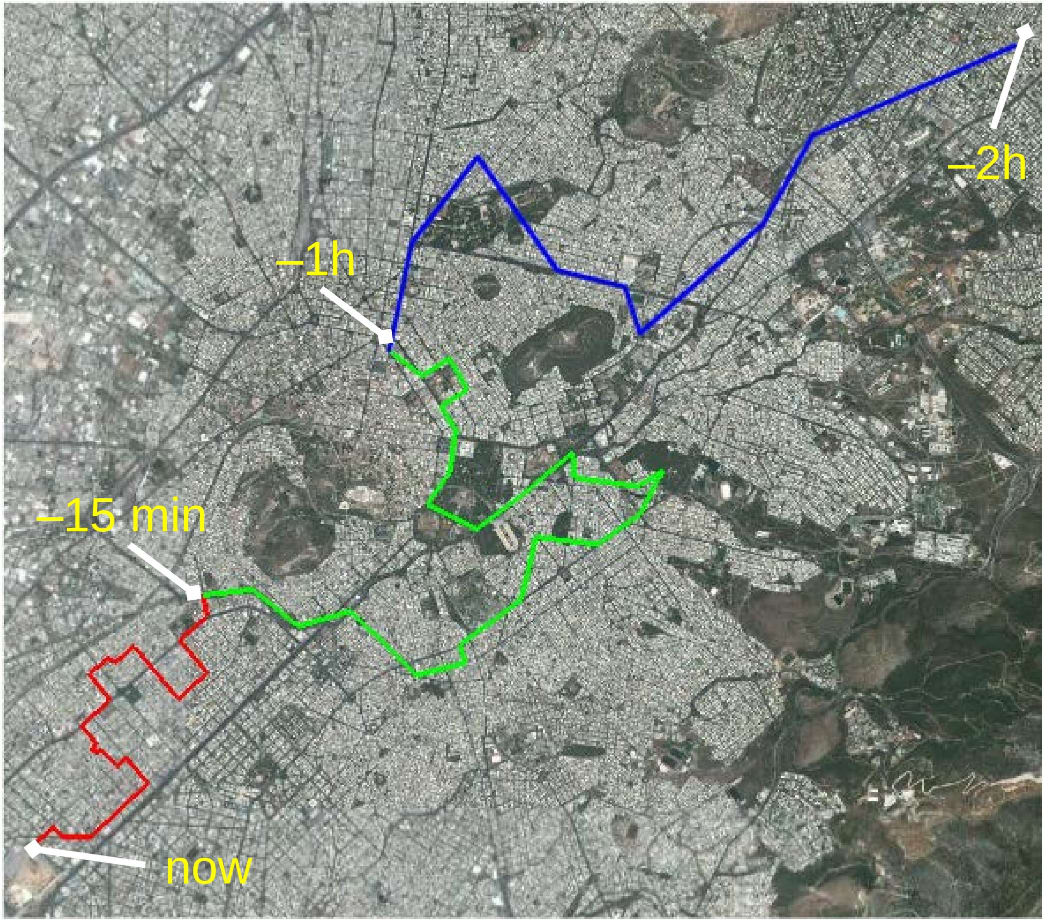
Example

- *Online simplifier per trajectory at prescribed resolutions & horizons*
- Multi-level window

RANGE 2 HOURS
SLIDE 15 MINUTES
SCALE 0.1

RANGE 1 HOUR
SLIDE 5 MINUTES
SCALE 0.3

RANGE 15 MINUTES
SLIDE 1 MINUTE
SCALE 0.5



Contribution

- Novel composite windows over trajectory streams
  - Capture essential motion features
    - Through multi-scale sliding windows
    - ➢ *With parameterized semantics in space and time*
  - Effectively "*drop detail with age*"
    - Through a user-defined "*scale*" factor
  - Ensure cohesion among trajectory segments
    - Maintenance methods for compressed representation
    - ➢ Designate *articulation points* that leave no gaps between levels
  - Specify continuous queries on trajectories
    - Applicability of typical spatiotemporal predicates
    - Expressiveness & clarity in query syntax

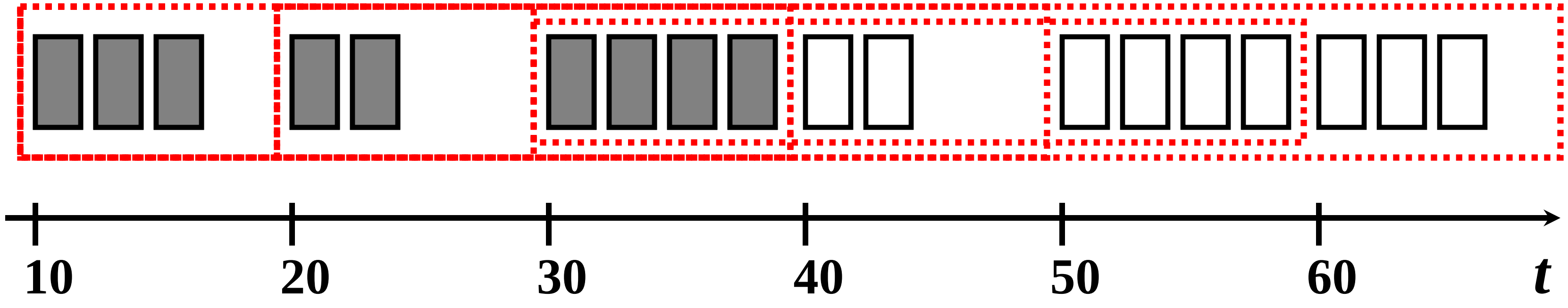
Outline

- Preliminaries: Sliding Windows *and beyond*
  - Window specifications
  - Multi-granular semantics
  - Trajectory management
- A Framework for Multi-scaling Windows
  - Model & Rationale
  - Semantics of *time-* & *trajectory-*based filtering
  - Properties
- Maintenance of Window States
  - Issues on shared & incremental computation
- Perspectives
  - Expressing windowed spatiotemporal queries
- Concluding Remarks

Time-based Sliding Windows

- Restrict stream items according to their timestamps
  - Fixed-size temporal scope ω [**RANGE:** *10 min*, *1h*,…]
    - A time interval against most recent stream items
  - Progression by slide step β [**SLIDE:** *1 sec*, *5 min*,…]
    - Stream items *expire* from the rear bound
    - *Possible overlaps*: common items between window states

➢ EXAMPLE: *window with* $\omega = 30, \beta = 10$ *time units*

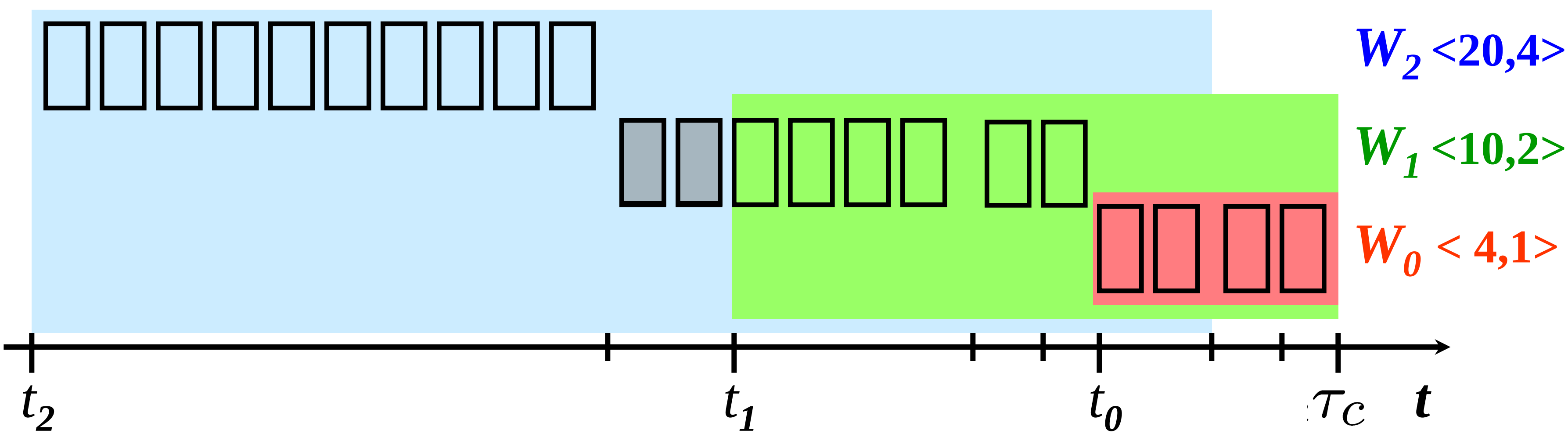


Multi-Granular Semantics

- Sliding window paradigm
  - Typically, a single timeline with all instants at similar detail
- Time implies multiple levels of resolution [LNCS 1399]
  - ➢ *seconds, minutes, hours, days*, weeks/months, years …
  - *granule* γ_k at level k: fixed number of timestamps in $\mathbb{T}$
    - A set of *contiguous timestamps* – no “holes” or “gaps”
  - *Consecutive granules* at level k → a granule at k+1
    - Iteratively define a hierarchy with multiple levels of granularity
- Also applied to spatiotemporal modeling
  - Multi-representations by scale, time, user, point of view [InfSys’06]
  - Multi-granular types, values, conversions, queries [ICOODB’09]
- Sliding windows at *subsumable time horizons* [TIME’10]
  - Effectively, each window frame represents a granularity level

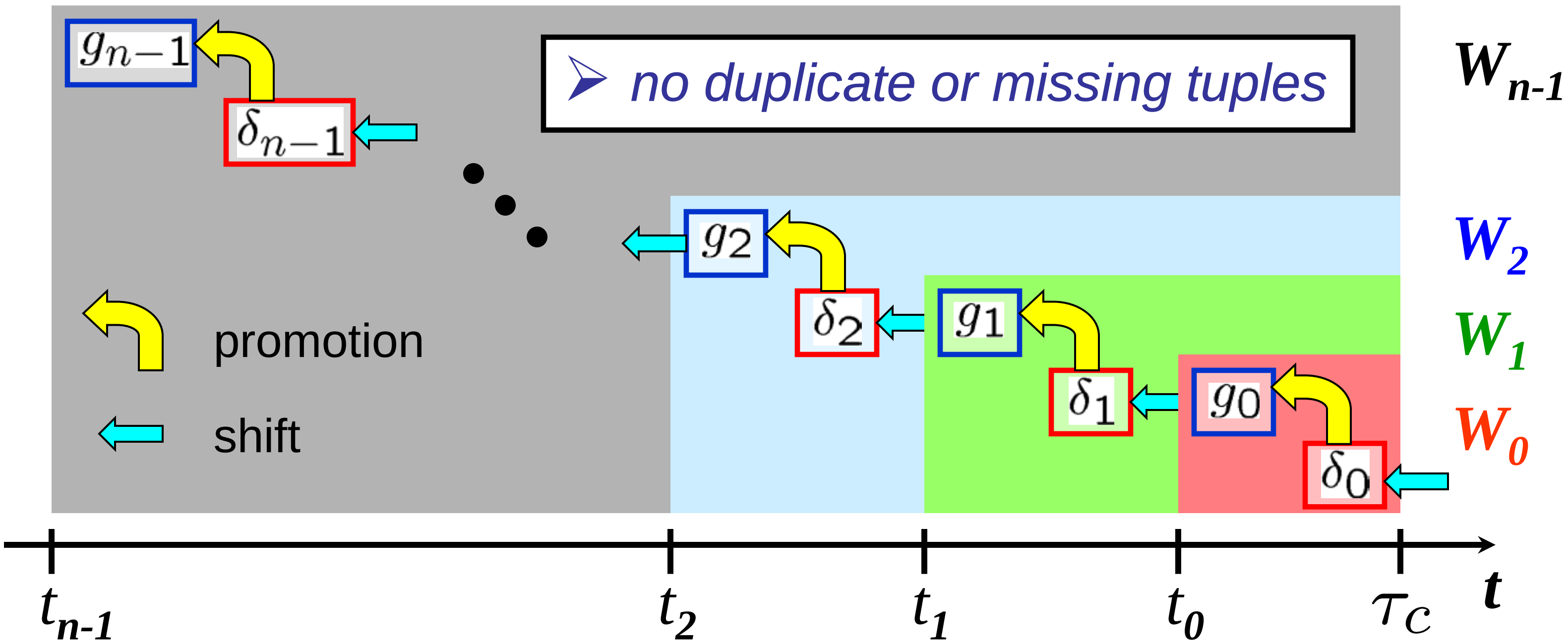
Multi-level Sliding Windows

- Window W with n levels of granularity [TIME'10]
  - At level k, subwindow W_k specifies a pair $\langle \omega_k, \beta_k \rangle$
    - **RANGE** ω_k *backwards* from current τ_C $\qquad \omega_{k-1} < \omega_k$
    - **SLIDE** *forward* periodically by β_k $\qquad \beta_{k-1} \leq \beta_k$
  - Hierarchy of nested frames *at multiple time horizons*
    - We assume *a fixed number of granules* per level, i.e., $\omega_k = \mu_k \cdot \beta_k$
  - *Issue:* overlapping ranges may *not be synchronized* at each τ_C



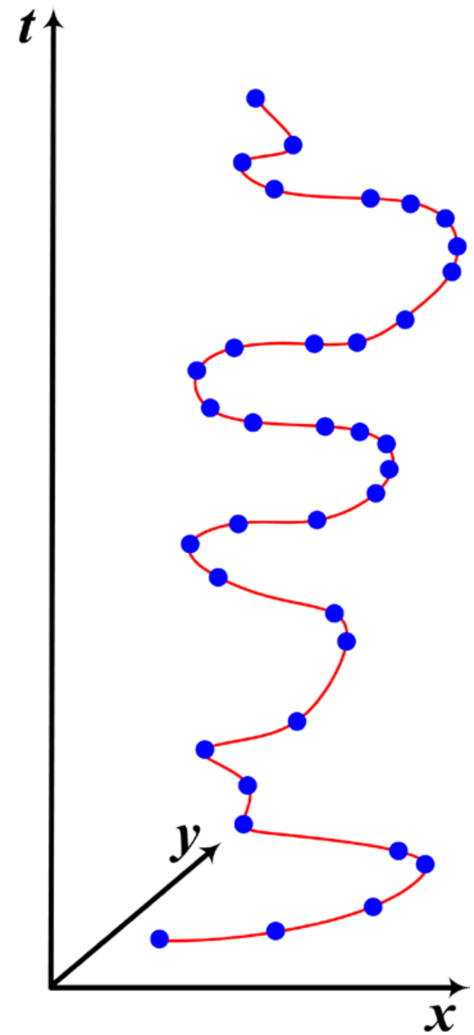
Stairwise Maintenance Scheme

- A "*flight of stairs*" forms a chain of *alternating*…
  - *Core nodes* g_k → queue items not covered by subordinate frames
  - *Auxiliary nodes* δ_k → buffer items expiring from a subwindow
  - ➢ Smooth maintenance: no tuples lost *in transit* between levels



Trajectory Modeling

- Monitor objects moving in a 3-*d* space
  - *Positional samples* are 2-*d* coordinates (*x*,*y*)
  - Measured at *discrete timestamps t*
    - Expressed in primitive units (e.g., seconds)
  - Each object with known identity *id*
- Massive updates from numerous objects
  - relayed into a server at varying frequency
  - ➢ a *positional stream* of tuples < *id*, *x*, *y*, *t* >
- Trajectory of a moving object
  - An evolving sequence of timestamped locations
    - an *append-only* collection of its GPS waypoints
    - *Ever growing traces of too much detail*…
  - ➢ Exploit *time monotonicity + windows*
    - to examine lightweight, but connected 3-*d* paths



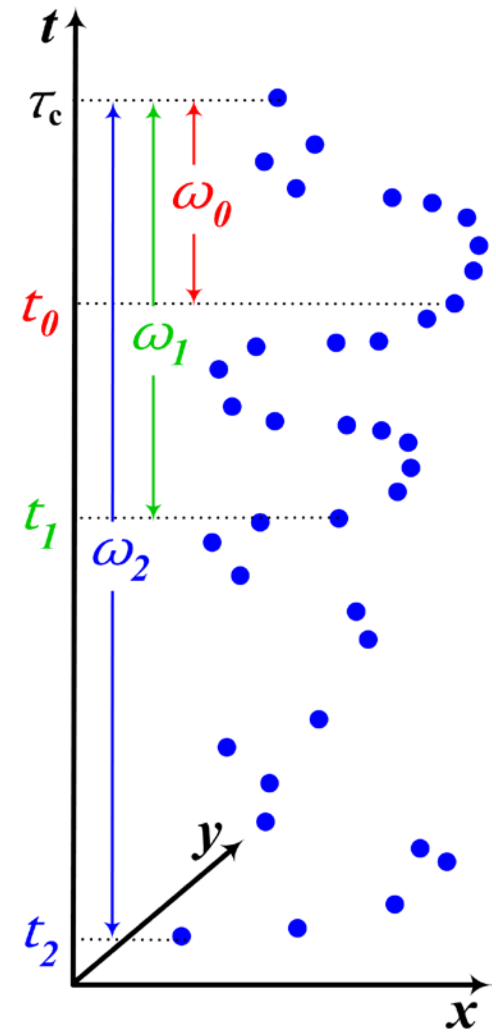
Multi-scale Window Semantics

- A window operator over streaming trajectories
  - *Time restriction* → focus on recent time periods only
  - *Data reduction* → progressively drop positions by age
  - ➢ Exploit spatial + temporal properties of trajectories
- Window is always specified by a continuous query
  - Evaluate CQ against less detailed object paths
  - *On-the-fly* trajectory compression at varying resolutions
- Operation: Filter object locations at two stages
  - *Time-based filtering*
    - Finite portions of reported locations at subsumable time intervals
  - *Trajectory-based filtering*
    - Group locations by object
    - Apply regulated generalization per trajectory at multiple scales

Time-based Filtering

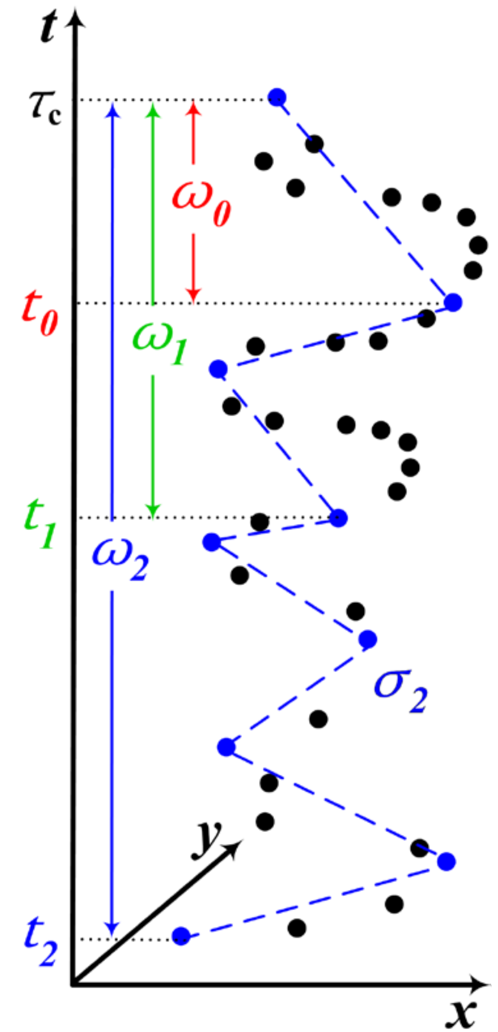
- Retain traces over diverse time horizons
  - Each window W_k has its own specs $\langle \omega_k, \beta_k \rangle$
  - All window frames initially applied at time τ_0
  - At any $\tau_c \geq \tau_0$, each W_k has a *scope*
    - i.e., the current window *bounds*

    $[\ \mathtt{max}\{\tau_0,\ \ \tau_c - \lambda_k - \omega_k + 1\},\ \ \tau_c - \lambda_k\ ]$
- Frames slide forward *discontinuously*
  - At a time-varying *lag* $\lambda_k = mod(\tau_c - \tau_0, \beta_k)$
  - Not at each timestamp or upon arrival of items
  - *Both bounds always move in tandem*
- $\mathcal{C}_k(\tau_c)$: *Time-filtered state* per frame W_k
  - i.e., positions measured during past periods
    - *last 15 min*, *1 hour*, *2 hours* etc.



Trajectory-based Filtering

- A *demultiplexing* phase per frame W_k
  - Partition items in $\mathcal{C}_k(\tau_c)$ into distinct paths
    - $path_k(i)$: time-ordered positions per object i
  - Still, filtered data may be considerable
    - for windows with large scopes or many levels
- Maybe diverse *reporting rate* ρ_i per object
  - $\rho_i \cdot \omega_k$ locations relayed during interval ω_k
  - ➢ Randomly discard *or* judiciously select samples
  - Let $\delta_k^i \cdot \omega_k$ locations remain *at most*
- $\sigma_k = \frac{\delta_k^i}{\rho_i}$: *reduction ratio* applied to $path_k(i)$
  - A common *scale* $\sigma_k < 1$ per frame W_k
    - Prescribes *max detail* tolerated at each path
  - A *reduced* $path'_k(i)$ is derived



Trajectory-filtered Window States

- Reduced paths do *not* have equal sample counts
  - Each trajectory is smoothed *separately*
    - but yields comparable representations
  - Derived paths are *connected* with no gaps
    - *Thanks to inherent timestamp ordering*
  - Approximate according to motion pattern & reporting frequency
    - e.g., discard superfluous points when moving along a straight path at constant speed
    - *Smaller scale* σ_k → *more approximation*
- *Trajectory-filtered state* per level W_k

  ➢ $\mathcal{W}_k(\tau_c) = \{path'_k(i), \forall\ object\ i\}$

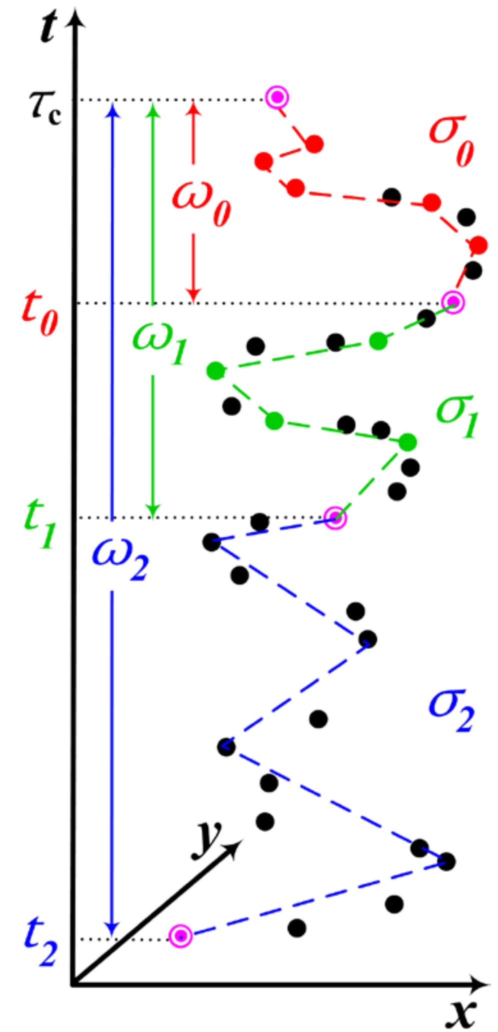
  - At each scale, one compressed sequence of samples per object
    - not just dispersed timestamped locations

Discussion

- Windowing applies *to query evaluation*, *not storage*
  - Original trajectories do *not* become multi-granular
    - *Granules*: levels of detail for reduced paths
    - Relationships among granularities are useful
    - "*finer-than*", "*coarser-than*" may simplify processing
  - Uniform, simple underlying data model
    - must include *position* + *timestamp*
  - ➢ Windows only provide *temporary paths* to queries
    - *User-defined* parameterization for **`RANGE, SLIDE, SCALE`**
  - Multi-scale windowing is *novel* in data stream processing
    - Distinguished from *partitioned windows*
    - Differs from *load shedding* policies
    - A "*path-creation*" step is employed to provide *sequential data*

Window Maintenance Issues

- Window-based trajectory simplification
  - A *repetitive task* due to evolving trajectories
    - *Incremental* → fresh positions are being relayed
    - *Shared* → reuse any paths already available
  - *Exploit point samples across window frames?*
    - More detailed representations closer to now
    - Fewer samples remain in coarser frames
    - ➢ Discard points upwards in window hierarchy
  - *Coordinated maintenance of multiple paths*
    - Preserve certain *articulation points* per trajectory
    - ➢ Pick samples temporally close to frame bounds
    - ➢ Minimize approximation error ~ *trajectory fitting*
  - Also a cohesive "*seamless*" *overview* of motion
    - non-overlapping paths joined at articulations



Trajectory Smoothing

- Alternatives for enforcing scale factor σ_k per frame W_k
  - Systematic sampling per trajectory
    - Start from most recent object position per frame
    - Randomly pick a sample from each successive batch of $\lceil \frac{\rho_i}{\delta_k^i} \rceil$ points
    - ➢ *Single-pass* method; may yield distorted paths
  - Minimal distance errors
    - Eliminate points incurring the least change in shape
    - Attain bounded space cost per level ($\sigma_k \cdot \omega_k$ locations)
    - ➢ *Multi-pass* algorithm; a Douglas-Peucker variant
  - Online filtering at frame transitions
    - Employ *stairwise scheme*
    - Only buffer nodes contain candidate positions for filtering
    - Drop samples when aging locations ascend through window hierarchy
    - ➢ *Inherent nesting*: no need to handle segments covered by lower frames

Windowed Queries on Trajectories

- Each window instantiation offers
  - An updated set of recent paths per monitored object
  - Contiguous traces → typical trajectories, *although approximate*
- Proposed *language constructs* to abstract data features
  - Functions returning "*timestamped polylines*"
    - **`trace()`**: reconstructs a separate path per subwindow
    - **`trajectory()`**: yields seamless synopsis from multi-scaled segments
    - ➢ Result is a sequence of locations per object
  - *Topological* operators can be applied:
    - ➢ **`INTERSECTS, WITHIN, CROSSES,`** …
  - *Spatiotemporal* functions more meaningful:
    - ➢ **`speed, duration, heading, distance,`** …
  - Concise *window clause* for expressiveness
    - combining **`RANGE, SLIDE, SCALE`** parameters for all levels

Usage of Language Constructs

- Assume a positional stream *S* **`<id, pos, t>`** from vehicles
  - *CQ1: Estimate average speed against varying time periods & scales*

```
SELECT AVG(speed(trace(S.pos))), WSCOPE(*)
FROM S [RANGES 1 HOUR,     15 MINUTES, 1 MINUTE
        SLIDES 5 MINUTES,   1 MINUTE,  10 SECONDS
        SCALES 0.1,         0.2,       0.5          BY S.id];
```

  - *CQ2: Indicate vehicles circulating in greater Athens recently*

```
SELECT S.id, duration(trajectory(S.pos))
FROM S [RANGES 30 MINUTES, 10 MINUTES, 1 MINUTE
        SLIDES 10 MINUTES,  1 MINUTE,  15 SECONDS
        SCALES 0.1,        0.2,        0.4          BY S.id],
     Cities C
WHERE trajectory(S.pos) WITHIN C.polygon
AND C.name='Athens';
```

Perspectives

- Applications
  - Fleet management
  - Traffic surveillance
  - Wildlife preservation
  - Merchandise monitoring
  - Maritime control
  - Soldier tracking …
- Operations
  - Trajectory filtering for range or *k*-NN search
    - lightweight, synchronized trajectories at comparable scales
  - Ageing synopses for trajectories
    - smoothly updated, gracefully compressed
  - Motion mining
    - trends and patterns at varying resolutions
  - Online multi-grained aggregates
    - speed, heading, … to analyze motion per trajectory
  - Advanced visualization on maps at diverse zoom levels

Conclusions

- Towards a foundation for a novel windowing operator
  - At multiple resolutions against streaming trajectory data
  - Semantics based on spatial + temporal properties
    - Not only restricting scope *on recent features*
    - But also progressively *dropping redundant details*
  - ➢ Opportunities for efficient shared evaluation
  - ➢ Useful in expressing spatiotemporal continuous queries
- Directions for further study
  - Incremental maintenance of multiple window states
    - *Strategies trading off performance vs. quality of approximation*
  - Verify scalability & robustness against workloads
    - *Experimentation on real / synthetic datasets*

Multi-scale Windowing over Trajectory Streams

Thank you!