Online Event Recognition from Moving Vehicles

E Tsilionis\textsuperscript{1}, N Koutroumanis\textsuperscript{2}, P Nikitopoulos\textsuperscript{2}, C Doulkeridis\textsuperscript{2} and A Artikis\textsuperscript{2,1}

\textsuperscript{1}NCSR “Demokritos”, Greece
\textsuperscript{2}University of Piraeus, Greece

http://cer.iit.demokritos.gr
http://www.datastories.org
Structure

- A two component online fleet management system.
- Effective integration of spatial reasoning with temporal reasoning for online CER.
- Evaluation on large, real-world, heterogeneous data.
System Architecture

Streaming GPS traces

Data enrichment
- Weather enrichment
- POIs enrichment

Data Analytics
- Composite Event Recognition
- Composite Event Patterns

Weather data
- POIs database

Composite event stream
Enrichment of Mobility Data with Weather and Points-of-interest
Weather Enrichment — How does it work?

Input: \((x, y, t)\)

1. Open appropriate file (based on time \(t\))
2. Retrieve values of weather attribute
3. Return weather attribute value for \((x, y)\)

NetCDF library

Disk

Memory

GRIB binary files
Points-of-interest Enrichment

- Practically a stream-based, parallel distance join
  - Based on spatial partitioning 2D space (in cells)
  - Cells assigned to worker nodes ($W_1, W_2, \ldots$)
  - Streaming data set: assigned to enclosed cell
  - Static data set: assigned to enclosed cell and replicated to cells at distance $\leq \theta$
Composite Event Recognition Overview

Simple Events
- happensAt(moving(V_1), T_1)
- happensAt(iceOnRoad(V_1), T_1)
- happensAt(abruptAcceleration(V_1), T_5)
- happensAt(stopped(V_1), T_{10})

Composite Event Definitions
- holdsFor(dangerousDriving(V_1) = true, [T_1, T_{10}])

Input ▶ Recognition ▶ Output ■

**Event Recognition System**

**Composite Events**
- initiatedAt(dangerousDriving(V) = true, T) ← happensAt(iceOnRoad(V), T), holdsAt(highSpeed(V) = true, T).
- terminatedAt(dangerousDriving(V) = true, T) ← happensAt(stopped(V), T).
Event Calculus

- A logic programming language for representing and reasoning about events and their effects.
- Key components:
  - event (typically instantaneous).
  - fluent: a property that may have different values at different points in time.
- Built-in representation of inertia:
  - $F = V$ holds at a particular time-point if $F = V$ has been initiated by an event at some earlier time-point, and not terminated by another event in the meantime.
Run-Time Event Calculus (RTEC)

<table>
<thead>
<tr>
<th>Predicate</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>happensAt($E, T$)</td>
<td>Event $E$ occurs at time $T$</td>
</tr>
<tr>
<td>initiatedAt($F = V, T$)</td>
<td>At time $T$ a period of time for which $F = V$ is initiated</td>
</tr>
<tr>
<td>terminatedAt($F = V, T$)</td>
<td>At time $T$ a period of time for which $F = V$ is terminated</td>
</tr>
<tr>
<td>holdsFor($F = V, I$)</td>
<td>$I$ is the list of the maximal intervals for which $F = V$ holds continuously</td>
</tr>
<tr>
<td>holdsAt($F = V, T$)</td>
<td>The value of fluent $F$ is $V$ at time $T$</td>
</tr>
</tbody>
</table>

![Diagram showing time intervals and a window \( \omega \)](image-url)
## Input — Output events

<table>
<thead>
<tr>
<th>Events</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>moving(V, S)</code></td>
<td>Vehicle $V$ is moving with a speed $S$</td>
</tr>
<tr>
<td><code>stopped(V)</code></td>
<td>Vehicle $V$ is not moving</td>
</tr>
<tr>
<td><code>abrupt Acceleration(V)</code></td>
<td>Vehicle $V$ accelerates abruptly</td>
</tr>
<tr>
<td><code>abrupt Deceleration(V)</code></td>
<td>Vehicle $V$ decelerates abruptly</td>
</tr>
<tr>
<td><code>abrupt Cornering(V)</code></td>
<td>Vehicle $V$ turns abruptly</td>
</tr>
<tr>
<td><code>fuelLevel(V, L)</code></td>
<td>The level of fuel in tank of vehicle $V$ is $L$</td>
</tr>
<tr>
<td><code>iceOnRoad(V)</code></td>
<td>Vehicle $V$ is moving in an icy road</td>
</tr>
<tr>
<td><code>closeToGas(V)</code></td>
<td>Vehicle $V$ is near a gas station</td>
</tr>
<tr>
<td><code>highSpeed(V)</code></td>
<td>Vehicle $V$ exceeds the user-specified speed limit</td>
</tr>
<tr>
<td><code>dangerous Driving(V)</code></td>
<td>Vehicle $V$ is potentially moving in a dangerous way</td>
</tr>
<tr>
<td><code>reFuelOpportunity(V)</code></td>
<td>There is refueling opportunity for vehicle $V$</td>
</tr>
</tbody>
</table>
Visualisation of a dangerousDriving activity

initiate dangerousDriving(V) iff
(abruptAcceleration(V) or
iceOnRoad(V)),
highSpeed(V).

terminate dangerousDriving(V) iff
end highSpeed(V) or
stopped(V).
Visualisation of a *reFuelOpportunity* activity

\[
\text{initiate } reFuelOpportunity(V) \text{ if} \\
\text{closeToGas}(V), \\
\text{highSpeed}(V), \\
\text{fuelLevel}(V, L), \\
\text{threshold}(V, \text{fuel}, V_{\text{tank}}), L < \frac{V_{\text{tank}}}{4}. \\
\text{terminate } reFuelOpportunity(V) \text{ if} \\
\text{fuelLevel}(V, L), \\
\text{threshold}(V, \text{fuel}, V_{\text{tank}}), L \geq \frac{V_{\text{tank}}}{4}. \\
\]
Experimental Setup

- 4M GPS traces of moving commercial vehicles (Vodafone Innovus), 1 month

- Data enrichment:
  - Weather information (frozen precipitation surface), 120 GRIB files, ≈ 7.4GB
  - Points of interest (gas stations), 140K
Experimental Results

Data enrichment component

![Execution time vs CPU Cores](chart1)

- **POI Enrichment**
- **Weather Enrichment**

![Throughput vs Distance θ](chart2)

- **POI Enrichment**

---

Track & Know
Experimental Results

CER component

- Avg Recognition Time (ms)
- Throughput (Thousand events / second)

For different window sizes (hours) and core counts (1 core, 2 cores, 4 cores, 8 cores).
Incremental Reasoning

- Handle delayed arrival of input events
- Use of overlapping temporal windows
- Avoid re-computation from scratch, i.e. RTEC
- Update only the CEs affected
- Improve the computational performance
Incremental Reasoning

Rule Transformation

\[
\text{initiatedAt}(\text{dangerousDriving}(V) = \text{true}, \ T) \leftarrow \\
\text{happensAt}(\text{abruptAcceleration}(V), \ T), \\
\text{holdsAt}(\text{highSpeed}(V) = \text{true}, \ T).
\]

\[
\text{initiatedAt}(\text{dangerousDriving}(V) = \text{true}, \ T) \leftarrow \\
\left[ \text{happensAt}(\text{abruptAcceleration}(V), \ T) \right]_{\text{ins}}^{Q_i}, \\
\left[ \text{holdsAt}(\text{highSpeed}(V) = \text{true}, \ T) \right]_{\text{ins}}^{Q_i}.
\]
initiatedAt(dangerousDriving(V)=true, T) ←

[happensAt(abruptAcceleration(V), T)]^{ins}_{Q_i}, [holdsAt(highSpeed(V)=true, T)]^{Qi}_{q_i - \omega}

ω
initiatedAt(\text{dangerousDriving}(V) = \text{true}, T) \leftarrow \\
[happensAt(abruptAccelerateVation(V), T)]_{\text{ins}} , \\
[holdsAt(highSpeed(V) = \text{true}, T)]_{Q_i} . \\
\omega \\
q_i - \omega \\
q_{i-1} \\
q_i
initiatedAt(dangerousDriving(V)=true, T) ←

[happensAt(abruptAcceleration(V), T)] \text{ins} \quad , \quad \text{happensAt(abruptAcceleration(V), T)}

[holdsAt(highSpeed(V)=true, T)] Q_i .

\text{initiatedAt(dangerousDriving(V)=true, T)} ←
Incremental Reasoning

Computation

\[
\text{initiatedAt}(\text{dangerousDriving}(V) = \text{true}, T) \leftarrow
\]

\[
\text{[happensAt}(\text{abruptAcceleration}(V), T)\text{]}^{\text{ins}} ,
\]

\[
\text{[holdsAt}(\text{highSpeed}(V) = \text{true}, T)\text{]}^{Q_i} .
\]

\[
q_i - \omega \quad \omega \quad q_{i-1} \quad q_i
\]
Experimental Results

Evaluation

(a) 5%

(b) 10%

(c) 20%
Summary

- Presented a stream reasoning online fleet management system.
- Evaluation illustrates the scalability of the system and its capacity to operate in real-time.
- Future work:
  - User evaluation.
  - Use of deadlines mechanism to deal with longer CE intervals (RTEC2).