NTNU
Kunnskap for en bedre verden

Online Semantic Trajectory Compression

Mats Aksnessæther
Jonas Rønning

Advisor:
Svein Erik Bratsberg

10.12.2025

Abstract

The rise of GPS-enabled devices, mobile internet connectivity, and deployment of vehicle location services has drastically increased the generation of vehicle trajectory data. These streams must be compressed online to reduce storage and transmission cost, and preserve semantically meaningful movement patterns.

This paper proposes HYSOC, a hybrid online semantic trajectory compression framework designed to address data volume, latency and semantic value in large-scale GPS trajectory streams. HYSOC integrates real-time behavioral STOP/MOVE segmentation with both geometric and network-based compression. This allows it to operate on raw GPS data or map-matched road-network trajectories. The architecture combines grid-indexed streaming segmentation, semantic abstraction of stop events, and two move-compression strategies, including a geometric-aware online simplifier and a network-semantic referential encoder for road-network data. To assess its effectiveness HYSOC will be benchmarked against offline baselines, using standardized metrics for information preservation, compression ratio, latency and memory footprint on realistic, large-scale trajectory datasets.

Contents

List of Figures

1 Introduction

GPS-equipped mobile devices are collecting enormous amounts of spatial and temporal information. The exponential increase in the amount of such trajectory data has caused significant problems [1]. While the data is a valuable asset for location-based services, it creates high storage and transmission costs [2].

The flood of data is characterized by two fundamental properties. First, the sheer volume of data is a vast strain on storage infrastructure and data transmission networks. Second, the data is redundant. Trajectory data, captured as a series of (x, y, t) positioning fixes, is highly spatiotemporally autocorrelated [3]. Consecutive fixes are co-located, with temporally neighboring fixes referring to similar positions in space. This redundancy presents a clear opportunity for data compression.

1.1 Motivation

The problem created by massive trajectory data is not limited to volume alone. The conceptual problem lies in the utility of the data. Raw trajectory data is semantically poor. Humans plan, perceive, and communicate movement not as a stream of (x, y, t) tuples, but as meaningful events and paths, such as "driving on street g" or "waiting at tram line #5" [3].

The generation of massive trajectory data creates a two-fold crisis for data management systems. First is the technical crisis of data volume and transmission cost. Second is the conceptual crisis of data utility. Raw trajectory data lacks human-interpretable meaning and is difficult to use within a larger context. Therefore, modern compression systems must not only reduce the data to solve the technical crisis, but also enrich the data to increase utility in larger contexts. This presents a fundamental transition from classical geometric compression, to a system that replaces redundant low-level data with high-level conceptual representations.

At the same time, the purpose of trajectory compression has experienced a critical paradigm shift. Earlier studies in the field primarily targeted offline compression, where an entire trajectory is compressed after all GPS points have been collected. This approach, focused on reducing storage costs, is not realistic for resource-constrained GPS-enabled devices [2] and imposes high communication overheads by requiring the full dataset to be transmitted before compression.

The field has evolved toward online compression, in which GPS points are compressed as they arrive in real-time [2]. This shift is not simply an optimization for efficiency but a fundamental enabler for a new class of modern applications [2]. The focus has moved from minimizing storage to minimizing latency for real-time services.

1.2 Thesis introduction: HYSOC

This thesis proposes the **Hybrid Online Semantic Compression System (HYSOC)**.
HYSOC is a novel, fully online framework designed to operate in real-time on network-constrained trajectory streams. It is the first system to integrate two complementary modes of semantic understanding: (1) real-time behavioral segmentation to identify semantically meaningful events like stops, and (2) real-time path-referential encoding to identify and compress semantically similar movement patterns. By hybridizing these approaches, HYSOC aims to provide a high-compression, semantically rich, and low-latency solution that surpasses the capabilities of current, specialized state-of-the-art systems.

1.3 Goals and contributions

The primary goal of this thesis is to design, implement and evaluate HYSOC, a novel hybrid framework that integrates online behavioral (stop/move) segmentation and compression for both network constrained and geometrically constrained trajectories. HYSOC will be benchmarked

against established offline methods to validate its performance. To evaluate its performance and usability these research questions will be used:

- **Research Question 1:** Can a streaming system perform real-time semantic behavioral segmentation while considering both processing latency and semantic accuracy?
- **Research Question 2:** Does HYSOC improve compression performance over established methods? Specifically, will compression ratio and semantic accuracy of HYSOC outperform a non-hybrid system that only uses behavioral segmentation or referential compression?
- **Research Question 3:** Does HYSOC achieve a better balance between compression ratio, information loss and processing latency than non-hybrid offline algorithms?

1.4 Outline

This remainder of this report will cover the following:

- Section 2 will cover the current methods used for trajectory compression and standardized performance metrics.
- Section 3 will cover the current state-of-the-art within online trajectory compression and justification of the research gap.
- Section 4 will present our planned model implementation and how the model will be evaluated.
- Section 5 will conclude the report and explain plans for future work

2 Trajectory Compression

2.1 Semantic Enrichment and Grounding

The theoretical foundation of this system relies on shifting the domain of trajectory representation from geometric space to semantic space. While geometric representation models movement as a continuous function in a Euclidean coordinate system, semantic representation models movement as discrete state transitions within a topological graph [3]. This transformation effectively filters out sensor noise and redundancy by enforcing the constraints of the underlying transportation infrastructure upon the raw data stream.

2.1.1 Semantic Trajectory Compression (STC)

Semantic Trajectory Compression (STC) is defined as the reduction of spatiotemporal data dimensionality through the exploitation of network constraints [3]. In this approach, a trajectory is not defined by a dense sequence of coordinate tuples (x, y, t), but by a sparse sequence of semantic reference points.

This approach assumes that network-constrained movement is highly autocorrelated, since a vehicle on a specific road segment has limited degrees of freedom. Consequently, the trajectory can be losslessly represented by storing only the events where the mobility channel changes (e.g., turning at an intersection or entering a new road segment) [3]. By retaining only these semantic transitions and their associated timestamps, the system discards the redundant intermediate fixes that occur while the object maintains a consistent state on a network edge.

2.1.2 Map-Matching via Hidden Markov Models

To transition from the geometric domain to the semantic domain, map matching serves as a probabilistic inference layer. This process projects raw, error-prone GPS observations onto the logical graph $G = \{V, E\}$, where V represents network nodes (intersections) and E represents edges (road segments) [4].

The standard theoretical model for this is the Hidden Markov Model (HMM). Within this framework, the map-matching problem is treated as a state estimation problem.

- Hidden States: The true road segments e_i are modeled as hidden states that cannot be observed directly.
- Observations: The GPS fixes o_t are treated as stochastic outputs generated by these hidden states.
- Probabilities: The most likely path is computed using Observation Probability (modeling the likelihood that a GPS point o_t was generated by segment e_i based on spatial proximity) and Transition Probability (modeling the likelihood of moving from segment e_i to e_j based on network topology and connectivity).

2.2 Behavioral Segmentation: The STOP/MOVE Model

While map matching provides the spatial context of movement, efficient compression requires understanding the behavioral semantics of the trajectory. Theoretical frameworks in trajectory data mining assume that movement is not a continuous stream of uniform importance, but rather a sequence of distinct behavioral episodes [5]. This STOP/MOVE model partitions a trajectory T into two mutually exclusive states:

- Stops (Events): Periods where the object is stationary or confined to a negligible spatial area for a significant duration. These represent meaningful activities (e.g., parking, deliveries, traffic congestion).
- Moves (Transitions): Periods of displacement between stops. These represent the travel component of the trajectory.

2.2.1 The Offline Archetype: STSS

The theoretical ideal for implementing STOP/MOVE segmentation is the Semantic-Based Trajectory Segmentation Simplification (STSS) framework [6]. STSS operates on a "Divide and Conquer" philosophy, processing the entire historical trajectory in batch mode to achieve global optimization.

The methodology proceeds in three distinct phases, which serve as the benchmark for behavioral analysis:

Extract (Density-Based Clustering) To identify Stop episodes, STSS utilizes density-based clustering rather than simple velocity thresholds [6]. Specifically, it employs an adapted version of OPTICS (Ordering Points to Identify the Clustering Structure) [7]. Unlike algorithms that require a global density threshold (e.g., DBSCAN), OPTICS identifies clusters of varying densities, allowing the system to distinguish different types of stops (e.g., brief traffic stop vs. long-term parking event).

Crucially, the distance metric in this context is modified to reflect the physical reality of movement. the distance between two point (p_i, p_j) is calculated not as the Euclidean distance, but as the

summation of the trajectory path between them [6]:

$$td(p_i, p_j) = \sum_{k=i}^{j-1} d(p_k, p_{k+1}) \tag{1}$$

This ensures that the clustering respects the temporal sequencing of the trajectory.

Divide (Partitioning) Based on the extracted clusters, the trajectory is partitioned into a sequence of alternating segments. This step creates a semantic definition of the journey, identifying where the behavior shifts from static to dynamic.

Conquer (Heterogeneous Simplification) Finally, STSS applies specific compression strategies tailored to the semantic nature of each segment.

- Stop Segment are aggressively simplified to a single representative coordinate (e.g., the original point nearest to the cluster centroid), eliminating localized sensor noise.
- Move Segments are simplified using network-constrained line generalization (e.g., Binary Line Generalization trees), preserving the geometry of the path.

Because STSS requires access to the full dataset to build the reachability plots for OPTICS, it is theoretically impossible to implement directly in a zero-latency streaming environment.

2.3 Online Constraints and Streaming Algorithms

While the STOP/MOVE model provides a robust semantic framework, implementing it in a real-time environment introduces severe computational constraints. Unlike offline algorithms (e.g., STSS), which operate on the complete historical dataset $T_{0...N}$, online systems must process an unbounded stream $S = \{p_0, p_1, ...\}$ where at time t_i, only points $p_{0...i}$ are available.

The constraint of zero look-ahead necessitates the replacement of global optimization algorithms with local, single-pass approximations. This thesis relies on two specific classes of streaming algorithms to overcome this limitation: grid indexing for behavioral segmentation and Referential Encoding for path compression.

2.3.1 The Latency-Accuracy Trade-off

The primary challenge in streaming analysis is the management of the Latency-Accuracy Trade-off. Detecting complex semantic events (such as STOP) typically requires buffering a significant window of historical points to distinguish transient traffic pauses from meaningful stationary behavior.

In a naive implementation, accuracy increases with buffer size, but computational latency increases linearly as the system iterates through the history to verify spatial density. To maintain real-time throughput, the system cannot rely on linear history scans. Instead it must utilize incremental maintenance, where the state of the trajectory is update in $O(1)$ time upon the arrival of each new point, regardless of the stream duration.

2.3.2 Streaming Segmentation: Grid Indexing

To approximate density-based clustering (like OPTICS) without the computational cost of global distance calculations ($O(N^2)$), streaming systems often utilize Grid Indexing [8]. This technique discretizes the continuous spatial domain into a matrix of cells. Incoming points are mapped to cell IDs rather than arbitrary coordinates. This allows the spatial history to be categorized into logical zones, such as confirmed clusters and pruned outliers, based on cell occupancy counts, reducing the search space for neighbor queries to a constant factor.

2.3.3 Referential Encoding via k-mer Matching

For path compression, Referential Encoding offers a semantic alternative to geometric simplification. Adapted from genomics, this technique decomposes a sequence of tokens (e.g., road segment IDs) into subsequences of length k, termed k-mers [2]. By maintaining a dynamic dictionary (hash table) of frequently observed k-mers, systems can replace raw sequence data with lightweight pointers $(ReferenceID, Offset)$. This approach exploits the "habitual" nature of urban traffic to achieve high compression ratios without requiring the geometric analysis of line simplification algorithms.

2.4 Performance Metrics

To objectively quantify the performance of trajectory compression algorithms, the literature relies on a standardized set of metrics. These metrics are categorized into two domains: Information Preservation (assessing the fidelity of the reconstructed data) and System Efficiency (assessing the computational cost) [9].

2.4.1 Accuracy Metrics (Information Preservation)

Standard geometric error metrics, such as Perpendicular Distance (PD), quantify spatial deviation but often neglect the temporal dimension of movement [9]. In trajectory analysis, preserving the spatiotemporal integration, the "when" alongside the "where", is essential for accurately reconstructing velocity profiles.

Synchronized Euclidean Distance (SED) The Synchronized Euclidean Distance (SED) is the standard metric for quantifying spatiotemporal error. Unlike spatial-only metrics, SED calculates the Euclidean distance between a point p_i in the original trajectory T and its temporally synchronized counterpart p'_i in the compressed representation T' [9].

Mathematically, for a compressed segment approximating motion between timestamps t_s and t_e, the synchronized position $p'_i(x'_i, y'_i)$ is derived via linear interpolation:

$$x'_i = x_s + \frac{t_i - t_s}{t_e - t_s}(x_e - x_s) \tag{2}$$

$$y'_i = y_s + \frac{t_i - t_s}{t_e - t_s}(y_e - y_s) \tag{3}$$

The SED is defined as the Euclidean distance between the actual position (x_i, y_i) and the interpolated position (x'_i, y'_i) [9]. This metric provides a rigorous assessment of how well an algorithm maintains the original speed and acceleration characteristics of the moving object.

Stop Detection F1-Score For systems performing behavioral segmentation (classifying as STOP or MOVE), accuracy is evaluated using standard information retrieval metrics. The $F_1 - Score$ is the harmonic mean of Precision and Recall, providing a single metric that balances Type I (False Positive) and Type II (False Negative) errors:

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

In this context, Precision measures the proportion of detected stops that correspond to actual stationary events, while Recall measures the proportion of actual stationary events successfully identified by the algorithm.

2.4.2 System Efficiency Metrics

Efficiency metrics quantify the computational resources required to perform the compression.

Compression Ratio (CR) The efficiency of data reduction is mathematically defined as the ratio of the uncompressed input size to the compressed output size:

$$CR = \frac{|Data_{original}|}{|Data_{compressed}|} \tag{5}$$

Higher values indicate greater efficiency. In semantic compression contexts, $|Data_{compressed}|$ must account for the storage overhead of semantic tokens (e.g., road IDs, timestamps) rather than simple coordinate tuples.

Computational Latency For streaming algorithms, efficiency is further defined by Processing Latency, typically expressed as the average time required to process a single data point ($\frac{\mu s}{point}$). Theoretically for a system to operate "online", this latency must remain lower than the data ingestion rate of the stream.

3 State of the Art

This section details the most advanced online semantic systems that constitute the direct competitors to HYSOC. This analysis establishes the research gap that HYSOC is designed to fill.

3.1 STEP

The STEP (Streaming Trajectory Segmentation framework based on stay Points) [10] is the state-of-the-art for online behavioral (stop) segmentation. It is explicitly designed to solve the latency-accuracy trade-off in processing streaming data [10].

STEP is a framework for streams, contrasting with STSS's offline approach [6]. The framework of STEP, visualized in Figure 1, consists of three modules (Indexing, Stay Point Detection, Trajectory Segmentation) designed to process "partial data" [10].

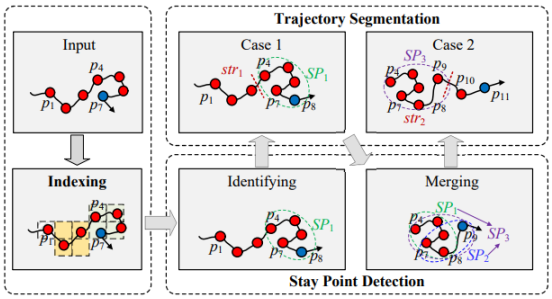

Figure 1: STEP Framework [10]

The core innovation of STEP lies in its well-designed grid index. As GPS points arrive in the stream, they are immediately placed into this grid structure. The primary objective is to identify a stay point, which requires checking if a point has remained within a specific distance `D` for a specific time `T`. Rather than scanning a long buffer of past points, STEP leverages its grid index for rapid querying [10]. This index uses a pruning strategy based on three defined areas: a Confirmed Area (points definitely within `D`), a Pruned Area (points definitely outside `D`), and a Check Area

(points that require exact calculation). This efficient, grid-based indexing allows STEP to perform stay-point detection using only recent data [10]. Once a segmentation is confirmed (either as a STOP (stay point) or a MOVE), it is assigned a "closed" status and immediately flushed from memory. This strategy allows the system to maintain a small memory footprint and low latency [10].

STEP represents the current state-of-the-art for the behavioral semantic component of trajectory analysis, but it does not address the necessary compression of the "move" segments themselves.

3.2 TRACE

The TRACE (online TRAjectory Compression) framework represents the state-of-the-art for online referential compression, specifically designed to find and compress subtrajectory similarities in real-time [2].

TRACE's core innovation is the adaptation of k-mer matching, a technique borrowed from genomics, to the challenge of trajectory data streams [2]. In this context, a "k-mer" is defined as a subsequence of a fixed length `k`. While in genomics a k-mer is a string of `k` base pairs, in TRACE it corresponds to a sequence of `k` road segments derived from the HMM-based map-matching output. The framework maintains a hash table `H` of previously encountered k-mers (subtrajectories), which collectively serve as a "reference set" [2].

When a new streaming trajectory arrives, the stream is immediately decomposed into k-mers. TRACE then calculates the hash key for the current k-mer and checks whether a matched subsequence exists in H [2]. If a match is found, the algorithm proceeds to greedy match the subsequence characters (i.e., road segments) until a definitive mismatch occurs [2]. The trajectory is then efficiently compressed by storing only a reference (such as a pointer `ref_id` and an offset `S`) to the existing subsequence in the reference set, instead of storing the entire sequence of road segments [2].

TRACE is a fully online framework, as evidenced by its online functions for reference deletion (to manage memory) and reference rewriting (to adapt to changing traffic patterns) [2]. While TRACE represents the state-of-the-art for the referential semantic component, it does not assign any behavioral meaning (like a "stop" or "move") to the trajectories it compresses.

3.3 SQUISH

SQUISH (Spatial QUality SImplification Heuristic) [1] is an online simplification method designed to approximate the goal of the offline Ramer-Douglas Peucker (DP) algorithm in an online streaming manner.

Its core principle is to use a fixed-size buffer, which acts as a sliding window, and a priority queue to decide which points to discard as new ones arrive. The priority of each point in the buffer is its estimated geometric error, calculating the local distance from that point to the line segment connecting its two immediate neighbors [1].

The logic visualized in Figure 2 proceeds as follows:

1. The algorithm maintains a fixed-size buffer (e.g., 100 points) implemented as a priority queue (min heap)
2. When the buffer is full and a new point arrives, the point with the lowest geometric error (the least important geometric point) is removed from the buffer.
3. The new point is then inserted, and the geometric errors for its two new neighbors are recomputed and their positions in the priority queue are updated.

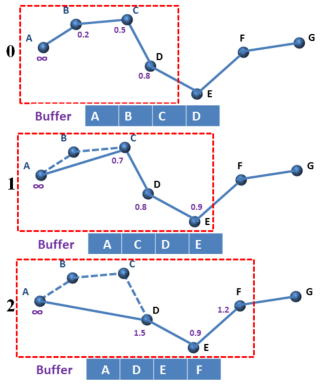

Figure 2: SQUISH algorithm with sliding window size and geometric updates [1]

SQUISH is a greedy, local optimization that effectively saves the most geometrically significant points within its buffer, making it suitable for high quality online simplification [1].

3.4 Research gap

This review of state-of-the-art systems reveals that the field of semantic compression has separated into two distinct, specialized, and non-overlapping subfields. The first is Behavioral Semantics (exemplified by STEP [10]), which focuses on events and answers the question, "What is the object doing?". Its primary semantic output is the identification and segmentation of a STOP event [10]. The second is Referential Semantics (exemplified by TRACE [2]), which focuses on patterns and answers the question, "Is this path common?". Its semantic output is a "reference" to a previously seen path, enabling compression [2].

The critical research gap is the absence of a system that integrates both approaches, a specialization that causes significant performance limitations. Systems relying solely on referential compression, such as TRACE [2], depend on matching speed and path data to historical references. This forces the system to process a 20-minute traffic jam continuously as a stream of speed updates rather than abstracting the event into a single semantic tuple. While segmentation-focused designs like STEP [10] successfully identify the 20-minute stop via stay-point detection, they introduce a different flaw. Because STEP treats the subsequent 10-kilometer "move" segment as a completed sub-trajectory to be flushed from memory, it fails to apply referential compression to minimize storage for that common commuting path.

The HYSOC framework proposed in this thesis is unique and necessary because it is the first hybrid system designed to integrate both of these online semantic models. HYSOC will function as an intelligent, hybrid framework by first processing the HMM stream through a behavioral module (inspired by STEP [10]) to detect and segment semantically rich STOP events. The remaining MOVE segments will then be passed to a referential module (inspired by TRACE [2]) to be compressed against a dynamic reference set of common paths. This hybrid approach allows HYSOC to achieve a higher, more holistic level of semantic compression than either specialized method can achieve alone, effectively solving both parts of the semantic challenge in real time.

4 Proposed system and methodology

The core contribution of this thesis is the Hybrid Semantic Online Compression (HYSOC) framework. HYSOC is a modular, real-time architecture designed to bridge the functional divide between behavioral and referential semantics identified in Section 3.

4.1 Introduction: The Imperative for Hybridization

While existing state-of-the-art systems are specialized to focus either on solely event detection (STEP) or path redundancy (TRACE), HYSOC will integrate these approaches into a single, unified pipeline. The framework adapts the "divide and conquer" philosophy of the offline STSS algorithm for a strict streaming environment. Unlike STSS, which relies on global knowledge of the trajectory, HYSOC operates without access to future data. It resolves the challenge of real-time semantic enrichment by replacing offline batch processing with high-performance streaming algorithms, ensuring both high compression ratios and immediate data utility.

4.2 The HYSOC Architecture

The HYSOC framework is structured as a strictly sequential streaming pipeline. The architecture accepts a raw stream of GPS coordinates and processes them through three distinct modules, each responsible for a specific layer of semantic extraction.

Module I: The Streaming Segmenter We position Module I as the cognitive core of the system, tasked with partitioning the continuous flow of coordinates into mutually exclusive STOP and MOVE segments. The primary challenge in this online context is the latency-accuracy trade-off. Accurately verifying a *stay point* typically requires buffering incoming points and performing repeated Euclidean distance calculations against the entire history. This approach creates a computational bottleneck that scales poorly as the buffer grows.

To resolve this and ensure real-time throughput, Module I will utilize a high-performance grid index structure rather than a linear buffer [10]. By mapping each new point immediately into a discretized spatial domain, we can instantly categorize prior points into distinct zones: a *Confirmed Area* where neighbors are accepted immediately, a *Pruned Area* where distant points are discarded, and a smaller *Check Area* where explicit distance calculations are necessary. This logic allows the module to function as a semantic router with near-constant processing speed, regardless of the buffer size.

Module II: the Stop Segment Compressor This module is designed to handle the specific characteristics of segments identified as Stops. In raw trajectory data, a stop event rarely looks like a single point. Instead it manifests as a dense, chaotic cloud of GPS readings caused by measurement noise rather than actual movement [5].

We will address this by applying a strategy of semantic abstraction. Rather than attempting to preserve the geometry of GPS error, we will compress the entire segment into a single, semantically rich representative tuple. Once Module I signals that the vehicle has resumed moving, Module II calculates the original point nearest to the centroid of the buffered stop segment. This process effectively filters out sensor noise, resulting in a single coordinate pair augmented with start and end timestamps. This method achieves extremely high compression ratios while providing a location reference that is often more accurate than a single raw point.

Module III: The Move Segment Compressors In Module III we will process the MOVE segments. Recognizing that real-world applications differ in their access to digital map infrastructure, we design two compression strategies to ensure versatility.

Strategy A: HYSOC-G (Geometric Implementation) For scenarios where map data is unavailable (network-agnostic), we integrate the SQUISH (Spatial Quality Simplification Heuristic) algorithm [1]. This strategy manages a dynamic buffer of trajectory points, prioritizing them based on their importance to the overall path [1]. We utilize the Synchronized Euclidean Distance (SED) metric rather than simple spatial distance [9]. SED accounts for the temporal dimension, ensuring that the compression preserves velocity changes and stops just as accurately as it preserves spatial

turns [9]. As the buffer fills, the algorithm discards the least critical points, effectively allocating the storage budget to the most significant spatiotemporal features [1].

Strategy B: HYSOC-N (Network-Semantic Implementation) HYSOC-N serves as our flagship implementation, designed to exploit the massive semantic redundancy inherent in road networks. This module functions as a two-stage pipeline. First, it employs a real-time Hidden Markov Model (HMM) to ground the noisy GPS points onto the road network [11], converting raw coordinates into a clean sequence of road segment identifiers. Second, these identifiers are fed into an online compression engine based on the TRACE framework [2]. By using k-mer matching to query a dynamic dictionary of common paths, this strategy replaces complex trajectory sequences with concise references, achieving maximal compression ratios.

4.3 Decompression and Reconstruction

To validate the utility of the compressed data stream and enable the calculation of error metrics, the HYSOC framework includes a decompression mechanism that logically inverts the operations of the compression modules. The reconstruction process is critical for restoring the trajectory geometry from the sparse compressed format.

The primary challenge lies in the reconstruction of the movement segments, which depends on the specific compression strategy employed. For the geometric implementation (HYSOC-G), the system applies Linear Interpolation between the preserved critical points. This interpolation is mathematically necessary to generate the synchronized points (p'_i) required to calculate the SED error metric defined in Equation (2). For the network-semantic implementation (HYSOC-N), the system must retrieve the original road segment identifiers (k-mers) associated with the stored Reference ID. This step restores the full sequence of road segments required for map matching and visualization.

4.4 The Benchmarking Framework: Offline Oracles

We evaluate HYSOC by comparing it against two offline benchmarks, referred to here as *Oracles*. Because these Oracles process data in batches rather than streams, they can look ahead to optimize compression in ways real-time systems cannot.

The initial processing step is identical for both Oracles. They process the full dataset using the STSS algorithm to separate stops from moves [6]. This is achieved using OPTICS clustering with a path-distance metric, which accurately identifies dense clusters of points along a route [7]. Once the data is segmented, the Oracles diverge in how they simplify the movement data.

The Geometric Oracle This benchmark validates the geometric efficiency of the system. It processes the "Move" segments using the Ramer-Douglas-Peucker (DP) algorithm [12]. Recognized as the standard for offline simplification, the DP algorithm utilizes global recursion to minimize the number of points required to define a shape, offering a strict target for our online geometric compressor.

The Network-Semantic Oracle This benchmark validates the semantic efficiency of the system. It processes the MOVE segments using Semantic Trajectory Compression [3]. This method groups continuous road segments into larger semantic chunks, reducing a long list of road IDs into a concise set of travel instructions. This represents the ideal output for a network-constrained compression system.

4.5 Datasets

Multiple publicly available trajectory datasets are suitable for evaluating online semantic compression methods. To validate the HYSOC framework, it is essential to have a dataset that reflects true scale and enough sampling to detect movement patterns. For the selection of datasets we have accounted for factors such as sampling rate, geographic coverage by OpenStreetMap (OSM), and availability for ground truth annotation for map-matching.

WorldTrace [13] is particularly well suited for our research. It provides pre-computed ground truth for OSM way identifiers, with matched coordinates and timestamps. WorldTrace covers 2.45 million trajectories from 70 countries collected between 2021 and 2023 [13]. Alternative datasets such as Chengdu and Xi´an taxi datasets [14] are viable because the underlying urban road network is well represented in OSM. A practical limitation is that many street names are stored in Chinese, which can make interpreting and debugging map-matching results more challenging for non-Chinese speakers. The widely used Porto taxi dataset provides an extensive set of of trajectories in a city excellently covered by OSM [15]. However, the dataset's sampling interval of 15 seconds is too coarse for detecting short stop-move segments and online compression in detail. For this reason, the thesis will prioritize datasets with higher resolution.

To evaluate the HYSOC framework we need to emulate live GPS collection. The selected datasets are processed by streaming individual trajectory points in chronological order by their timestamp. In this setup, each point is processed as soon as it arrives, and segmentation and compression are performed only on past and current information. This allows the HYSOC method to be tested under realistic conditions.

4.6 Evaluation Protocol and Metrics

To address the Research Questions defined in Section 1.3, we establish a rigorous evaluation protocol. While Section 2.4 defined the mathematical properties of the selected metrics, this section details their specific application within our experimental framework.

4.6.1 Evaluating Semantic Segmentation (RQ1)

To validate RQ1 (Real-time behavioral segmentation), we measure the ability of Module I to correctly identify STOP events compared to the synthetic ground truth generated by the offline STSS Oracle.

We utilize the $F_1 - Score$ as the primary indicator. However, quantify a match in continuous time requires a strict intersection criterion. A detected stop segment $S_{det} = [t_{start}, t_{end}]$ is classified as a True Positive (TP) if and only if it temporally overlaps with a ground-truth stop S_{gt} by at least 50% of its duration:

$$\frac{duration(S_{det} \cap S_{gt})}{duration(S_{det} \cup S_{gt})} \geq 0.5 \tag{6}$$

Segments failing this threshold constitute False Positives (FP), and missed ground-truth stops constitute False Negatives (FN). This strict criterion ensures that HYSOC is penalized for identifying stops that are merely traffic delays or GPS jitter.

4.6.2 Evaluating Compression Efficiency (RQ2)

To address RQ2 (Compression performance), we compare the file sizes of the HYSOC output against both the raw input and the non-hybrid baselines (STEP and TRACE).

We report the Compression Ratio (CR). For the semantic implementation (HYSOC-N), the output size is calculated as the sum of all stored reference tuples $(RefID, Offset)$ and semantic event

tuples $(Lat, Lon, t_{start}, t_{end})$. We specifically analyze the marginal gain of hybridization by comparing the CR of HYSOC against a system using only Module I (Stops only) or only Module III (Referential only).

4.6.3 Evaluating Information Loss (RQ3)

To answer RQ3 (Comparison against offline oracles), we quantify the distortion introduced by the online constraints.

Geometric Fidelity: We calculate the Synchronized Euclidean Distance (SED) for every point in the Move segments. Unlike offline algorithms that can optimize the entire path globally (e.g., Douglas-Peucker), HYSOC makes greedy decisions. We report the Mean SED and 95th Percentile SED to highlight worst-case deviations. A low SED confirms that HYSOC preserves the velocity profile required for travel time estimation.

Storage Overhead: Distinct from the original compression ratio, we measure the Peak Memory Footprint during processing. We log the maximum size of the Grid Index (Module I) and the Dictionary Hash Map (Module III) to verify that the flush mechanism successfully prevents memory drift over long streams.

4.6.4 Evaluating Throughput

To ensure the system meets the Online requirement, we measure Processing Latency. We stream the dataset at maximum speed and log the wall-clock time required to process each point ($\frac{\mu s}{point}$). The success criterion is a throughput that exceeds the sampling rate of the standard GPS devices (typically 1Hz), ensuring no back-pressure builds up in the ingestion buffer.

Finally, we must validate the computational cost of reconstruction. While compression latency dictates ingestion capacity, Decompression Latency determines the feasibility of real-time visualization. Therefore, we will measure the decompression throughput (points reconstructed per second) for both HYSOC-G and HYSOC-N. This metric is critical to ensure that hybrid overhead—specifically dictionary lookups—does not bottleneck the end-user data stream.

5 Conclusion

This report has explored the need for online semantic trajectory compression in the context of large-scale GPS data streams and the two-fold crisis of data volume and interpretability of raw GPS points. We have reviewed the theoretical foundations of semantic trajectory compression, behavioral segmentation, geometric and network-based compression together with standard metrics for evaluation. Additionally, the report has explored the state-of-the-art online systems for behavioral segmentation (STEP [10]), referential semantics (TRACE [2]), and geometric simplification (SQUISH [1]), and identified a research gap between these specialized approaches. Lastly, we have proposed the HYSOC architecture, which combines the different methods identified in the state-of-the-art research, together with a plan for evaluation against offline baselines.

5.1 Future work

The work presented in this report forms the conceptual and methodological foundation for our Master's thesis. The next step is the implementation of HYSOC, and its respective modules. Based on the evaluation protocol defined, the full thesis will then benchmark HYSOC against the offline and online baselines using large-scale trajectory datasets. Throughout the evaluation of HYSOC, the focus will be on quantifying compression ratio, information loss, latency and memory footprint. The implementation and experimental analysis in the Master's thesis will test the architecture and validate the design proposed in this specialization project.

Acknowledgments

We want to thank our supervisor Svein Erik Bratsberg for feedback and help throughout the process.

Bibliography

[1] Jonathan Muckell et al. 'Squish: An online approach for gps trajectory compression'. In: *Com.Geo 2011* (Jan. 2011), 13:1–13:8.

[2] Tianyi Li et al. 'TRACE: Real-time Compression of Streaming Trajectories in Road Networks'. In: *Proceedings of the VLDB Endowment* 14.7 (2021), pp. 1175–1187. DOI: 10.14778/3457396.3457398.

[3] Kai-Florian Richter, Falko Schmid and Patrick Laube. 'Semantic Trajectory Compression – Representing Urban Movement in a Nutshell'. In: *Journal of Spatial Information Science* 4 (June 2012), pp. 3–30. DOI: 10.5311/JOSIS.2012.4.62.

[4] Yingxue Zhang, Haowen Yan and Xiaomin Lu. 'An enhanced HMM map matching algorithm incorporating personal road selection preferences'. In: *Scientific Reports* 15 (Oct. 2025). DOI: 10.1038/s41598-025-14050-8.

[5] Zhixian Yan et al. 'SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories'. In: *Proceedings of the 14th International Conference on Extending Database Technology (EDBT 2011)*. ACM, 2011, pp. 259–270. DOI: 10.1145/1951365.1951398.

[6] Minshi Liu, Guifang He and Yi Long. 'A Semantics-Based Trajectory Segmentation Simplification Method'. In: *Journal of Geovisualization and Spatial Analysis* 5 (Dec. 2021). DOI: 10.1007/s41651-021-00088-5.

[7] Mihael Ankerst et al. 'OPTICS: Ordering Points to Identify the Clustering Structure'. In: *Sigmod Record* 28 (June 1999), pp. 49–60. DOI: 10.1145/304182.304187.

[8] Amineh Amini, Teh Ying Wah and Hadi Saboohi. 'On Density-Based Data Streams Clustering Algorithms: A Survey'. In: *Journal of Computer Science and Technology* 29.1 (2014), pp. 116–141. DOI: 10.1007/s11390-013-1416-3.

[9] Jonathan Muckell et al. *Compression of Trajectory Data: A Comprehensive Evaluation and New Approach*. Tech. rep. TR-SUNYA-CS-12-05. University at Albany, SUNY, 2013. URL: http://www.cs.albany.edu/~jhh/publications/TR-SUNYA-CS-12-05.pdf.

[10] Yangyang Sun et al. 'Streaming Trajectory Segmentation Based on Stay-Point Detection'. In: Springer-Verlag, Berlin, Heidelberg, Oct. 2024, pp. 203–213. ISBN: 978-981-97-5551-6. DOI: 10.1007/978-981-97-5552-3_13.

[11] Sinan Goh et al. 'Online Map-Matching Based on Hidden Markov Model for Real-Time Traffic Sensing Applications'. In: *2012 15th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2012, pp. 776–781. DOI: 10.1109/ITSC.2012.6338627.

[12] David H. Douglas and Thomas K. Peucker. 'Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature'. In: *Cartographica: The International Journal for Geographic Information and Geovisualization* 10.2 (1973), pp. 112–122. DOI: 10.3138/FM57-6770-U75U-7727.

[13] Yuanshao Zhu et al. 'UniTraj: Universal Human Trajectory Modeling from Billion-Scale Worldwide Traces'. In: (Nov. 2024). DOI: 10.48550/arXiv.2411.03859.

[14] Didi Chuxing. *GAIA Open Data Initiative: Trajectory and Road Network Dataset (Kaggle Mirror)*. Kaggle. Mirrored from the GAIA platform. Original provider: Didi Chuxing – GAIA Initiative. Accessed: 27 November 2025. n.d. URL: https://www.kaggle.com/datasets/ash1971/didi-dataset.

[15] Meghan O'Connell, moreiraMatias and Wendy Kan. *ECML/PKDD 15: Taxi Trajectory Prediction (I)*. https://kaggle.com/competitions/pkdd-15-predict-taxi-service-trajectory-i. Kaggle. 2015.