# Online Location Trajectory Compression

**ABM Musa** *
Amazon
Bellevue, WA 98004
musaabm@amazon.com

**James Biagioni** *
Carmera
Seattle, WA 98195
james@carmera.com

**Jakob Eriksson**
University of Illinois at Chicago
Chicago, IL 60607
jakob@uic.edu

## Abstract

This paper presents a system for online GPS tracking, where a device reports its location in near real-time to a central server over a cellular uplink. Here, a user can specify the error and the delay bound, and the system optimizes the uplink usage. Experiments show that this system reduces the data usage 20× or more compared to the status quo while providing improved guarantees and flexibility.

***K*eywords** trajectory compression · GPS tracking · online tracking

## 1 Introduction

Tracking assets and people using the global positioning system is widespread today. These applications include freight logistics, public transit arrival time prediction, anti-theft lowjack devices, and crowd-sourced traffic data collection using smartphone apps.

A typical online GPS tracking system uses a cellular uplink to report the location of a device to a central server. The general practice in online tracking today is to use periodic fixed-interval transmission where a small transmission interval is used if the data budget is high and vice-versa. However, there is ample room for improvement to the status quo in online tracking. This paper describes a system for efficient online GPS tracking with a given error and delay bound.

## 2 Overview

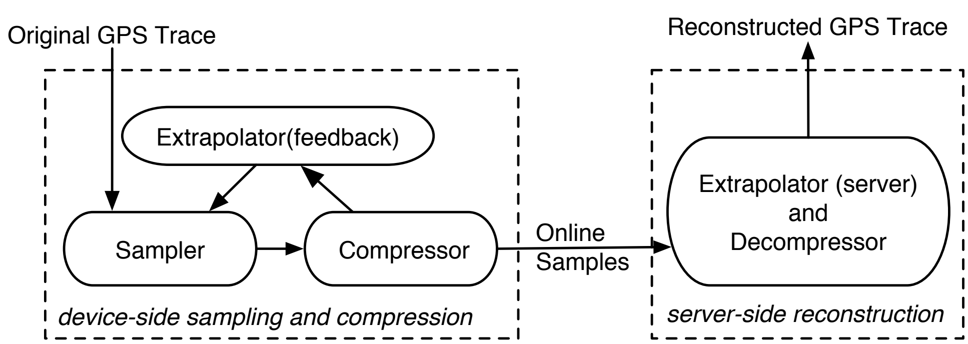


Figure 1: System Architecture.

Figure 1 illustrates the online tracking architecture. The incoming "raw" GPS trace frequency is typically 1 Hz. This trace is passed through a sampler, which decides whether to forward a given trace point to the server based on the error bound. At the same time, two identical extrapolators are run: one on the server, and another on the mobile device. An extrapolator produces a continuous location estimate at the current time starting from the last transmitted location. If a

*This work is done when the author was a PhD candidate at the University of Illinois at Chicago

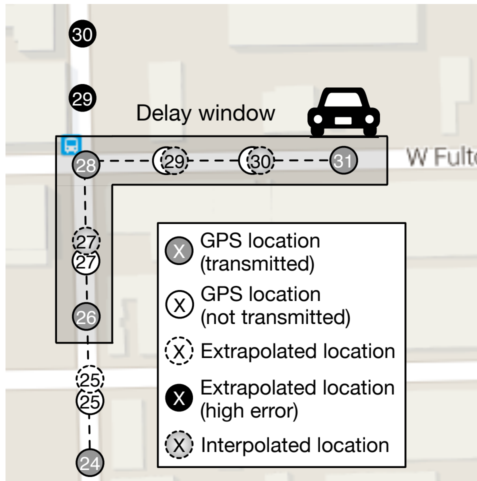


Figure 2: By introducing a fixed delay, a sampler can send multiple samples in one packet, and may choose more effectively what samples to send.

transmission is made, both extrapolators are updated On the server side, the extrapolator output is made available for use by the trace consumer. On the client side, an identical extrapolator produces a continuous location estimate for local use. By comparing the output of the local extrapolator against the incoming raw GPS location, the error-aware sampler makes its forwarding decision based on the difference between the current estimate and the measured location. If a fixed delay in reporting is allowed, the sampler can also flexibly choose when to transmit a sample and apply GPS trace compression. All operations here are lightweight and do not add much computational burden on the client and server.

## 3 Extrapolation

By estimating the location of a device without additional transmissions, extrapolation naturally helps improve accuracy and timeliness at the server. However, extrapolation can also be used to improve efficiency at the sender. Here, the sender replicates the extrapolation process performed at the receiver, enabling it to directly observe the extrapolation error incurred at the server. This, in turn, allows the sender to choose the samples it transmits to maximize the accuracy gained from each transmission.

The most basic extrapolation method is "Constant Location (CL)" that predicts the future location, for all times in the future, will be the same as the most recently reported location. This extrapolation method improves the timeliness of tracking since the system can provide an immediate estimate, at any point in time. However, this gain in timeliness is matched by a loss in accuracy: for a moving device, predictions made by this extrapolator grow increasingly inaccurate with the time since the last update.

Since a device to be tracked is moving most of the time, "Constant Velocity (CV)" extrapolator provides better estimates and this paper present results with CV extrapolator. However, it is possible to use more advanced extrapolators as well as combination of extrapolators for improved performance, which is discussed in [1, 2].

## 4 Sampler

With a maximum error bound and fixed delay configured by the user, the task of the sampler is to minimize mean data usage while enforcing the maximum error bound. For each incoming location from the GPS, the sampler measures the distance between the extrapolated trace and the current location. If the distance exceeds the maximum error bound, this sample must be transmitted. If zero delay is configured, the sample is transmitted immediately, updating the server and restarting the extrapolation. With a non-zero delay of $T$ seconds, the sample (and the surrounding window of samples) can be transmitted at once and GPS compression (see §6) can be used.

## 5 Delay

For maximum timeliness, a sampler must decide whether or not to transmit each sample as soon as it arrives, allowing relatively little room for maximizing sampling efficiency. However, if the user is willing to tolerate a fixed delay in the reporting, two additional gains in performance can be achieved. First, the sampler can have a better decision of which points to transmit. Because of the delay, the sampler can essentially look ahead and see if significant extrapolation errors will occur because of not transmitting some samples. Thus by transmitting such samples, the sampler can mitigate the high errors before they occur. Fig 2 shows an example of this. Here, the solid disks (white or gray) are actual locations, the dotted white disks are extrapolated locations, the black disks are extrapolated locations (not transmitted), and the gray disks are transmitted locations. In this example, The subject takes a right turn in the intersection. Here, significant extrapolation errors (black disks) are made before the turn because of failure to predict the turn, but with the added delay the sampler can capture this and avoid the extrapolation errors. Second, the sampler can apply GPS compression and transmit only a few key locations (gray disks). Then the server fills up for non-transmitted locations by interpolation (dotted disks with radial gray).

## 6 GPS Compression

GPS compression can yield substantial data usage savings with minimal accuracy loss, particularly for long delays. For error bound GPS compression, we need to select a subset of the GPS samples from the original trajectory so that during the reconstruction the error for each of those original samples is bounded by the given error bound. This paper uses the compression scheme proposed by [3]. Their technique is greedy and similar to the line compression algorithm by Douglas-Peuker [4]. The algorithm starts with two end samples and the reconstructed trajectory is simply an edge in that case. At each subsequent step, this algorithm selects a sample that reduces the error for the approximated trajectory the most and as a result, introduces a new edge. This selection process continues independently for all edges recursively until the errors for all original samples are below the given error bound. Here, time synchronized distance measure is used for the error metric as GPS trajectories are spatio-temporal in nature.

## 7 Results

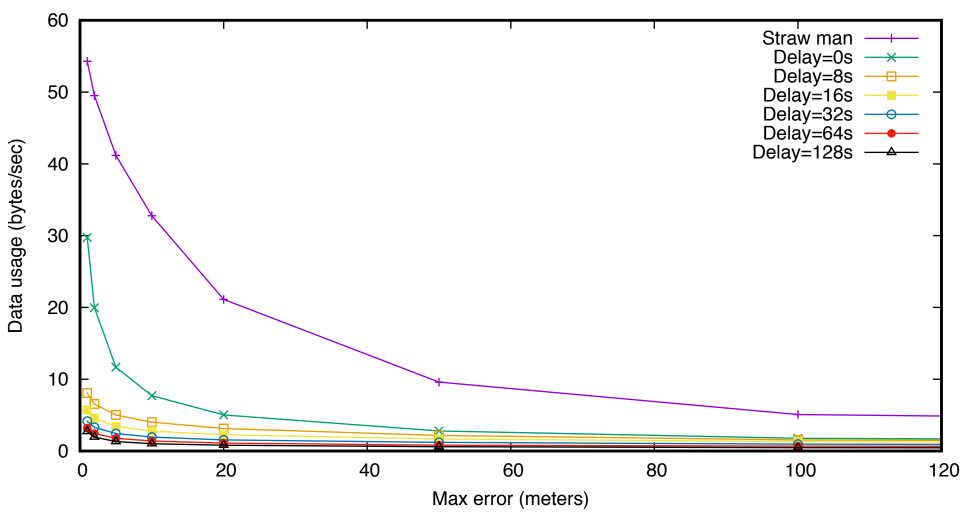


Figure 3: Uplink data usage with increasing maximum error bound, for usage optimizing sampler with various configured delays.

This system is evaluated using GPS traces obtained from OpenStreetMap (OSM). The total duration of this dataset is 3450 hours and it contains 12.4 million GPS points. Figure 3 shows the data usage of the system with constant velocity extrapolator and GPS compression as the maximum error threshold is varied, for different delays. The straw-man solution transmits a single sample with a fixed distance interval equal to the maximum error threshold, thus guaranteeing that the error never goes beyond the user provided error threshold. While the straw-man provides a guaranteed maximum error bound, it does so at high uplink usage. However, as one might expect, this data usage decreases with increasing maximum error tolerance. This system outperforms the straw-man by a significant margin even for zero delay due to extrapolation. As more delays are allowed, the data usage reduces significantly more.

Table 1 shows the percentage of data-usage reduction by this system compared to the straw-man solution focusing on the case of 20 m maximum error bound. For example, this system achieves a 76% reduction in data usage compared to the straw-man, with no delay. If the operator is willing to accept some delay, this system reduces data usage relative to the straw-man by 89% (9×) for 16 seconds delay and 95% (20×) for 64 seconds delay.

| **Delay** | **0s** | **8s** | **16s** | **32s** | **64s** | **128s** |
|---|---|---|---|---|---|---|
| Reduction (%) | 76% | 85% | 89% | 92% | 95% | 96% |

Table 1: Data usage reduction compared to straw-man with various delays (error=20 m)

## References


[1] ABM Musa, J. Biagioni, and J. Eriksson. Trading off accuracy, timeliness, and uplink usage in online gps tracking. *IEEE Transactions on Mobile Computing*, 15(8):2124–2136, Aug 2016.

[2] James Biagioni, ABM Musa, and Jakob Eriksson. Thrifty tracking: Online gps tracking with low data uplink usage. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'13, pages 496–499, New York, NY, USA, 2013. ACM.

[3] Nirvana Meratnia and A Rolf. Spatiotemporal compression techniques for moving point objects. In *EDBT*, pages 765–782. Springer, 2004.

[4] 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. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10, 1973.