Simplification of Trajectory Streams*

Siu-Wing Cheng† Haoqiang Huang† Le Jiang‡

The pervasive use of GPS sensors has enabled tracking of moving objects. For example, a car fleet can use the real-time information about its vehicles to deploy them in response to dynamic demands. The sensor periodically samples the location of the moving object and sends it as a vertex to the remote cloud server for storage and processing. The remote server interprets the sequence of vertices received as a polygonal curve (trajectory).

For a massive stream, sending all vertices uses too much network bandwidth and storage at the server, and it may aggravate issues like out-of-order and duplicate data points. Software systems have been built for the sensor side to simplify trajectory streams on the fly. A local buffer is used. Every incoming vertex in the stream triggers a new round of computation that uses only the incoming vertex and the data in the local buffer. At the end of a round, these systems may determine the next vertex and send it to the cloud server, but they may also send nothing.

It is important to keep the local buffer small. It means a small working storage and less computation. Let τ be the curve in the stream. The abovementioned software systems only deal with some local error measures between the simplified curve σ and τ. We want to provide guarantees on the size of σ and the global similarity between σ and τ.

A popular similarity measure is the *Fréchet distance*. Let $\rho_\tau : [0,1] \to \mathbb{R}^d$ be a *parameterization* of τ such that, as t increases from 0 to 1, $\rho_\tau(t)$ moves from the beginning of τ to its end without backtracking. It is possible that $\rho_\tau(t) = \rho_\tau(t')$ for two distinct $t, t' \in [0,1]$. We can similarly define a parameterization ρ_σ for σ. These parameterizations induce a *matching* $\mathcal{M}$ between $\rho_\sigma(t)$ and $\rho_\tau(t)$ for all $t \in [0,1]$. A point may have multiple matching partners. The Fréchet distance is $d_F(\sigma,\tau) = \inf_{\mathcal{M}} \max_{t\in[0,1]} d(\rho_\sigma(t), \rho_\tau(t))$, where $d(\cdot\,,\cdot)$ is the Eculidean distance.

We study two problems. The *δ-simplification problem* is to compute, for a given $\delta > 0$, a curve σ of the minimum size (number of vertices) such that $d_F(\sigma,\tau) \le \delta$. The *$k$-simplification problem* is to compute, for a given integer $k \ge 2$, a curve σ of size k that minimizes $d_F(\sigma,\tau)$.

For any $\varepsilon \in (0,1)$ and any $\delta > 0$, our δ-simplification algorithm produces a curve σ for the prefix $\tau[v_1, v_i]$ in the stream so far such that $d_F(\sigma, \tau[v_1,v_i]) \le (1+\varepsilon)\delta$ and $|\sigma| \le 2\min\{|\sigma'| : d_F(\tau[v_1,v_i],\sigma) \le \delta\} - 2$. Let $\alpha = 2(d-1)\lfloor d/2 \rfloor^2 + d$. The working storage is $O(\varepsilon^{-\alpha})$. Each vertex in the stream is processed in $O(\varepsilon^{-\alpha}\log\frac{1}{\varepsilon})$ time for $d \in \{2,3\}$ and $O(\varepsilon^{-\alpha})$ time for $d \ge 4$. If our algorithm is used in the static case, the running time on a curve of size n is $O(\varepsilon^{-\alpha} n \log\frac{1}{\varepsilon})$. Ignoring polynomial factors in $1/\varepsilon$, our time bound is a factor n smaller than the $O(\varepsilon^{2-2d} n^2 \log n \log\log n)$ running time of the best static algorithm that achieves the same error and size bounds.

For any $k \ge 2$ and any $\varepsilon \in (0, \frac{1}{17})$, our k-simplification algorithm guarantees that $|\sigma| \le 2k-2$ and $d_F(\sigma, \tau[v_1,v_i]) \le (1+\varepsilon)\cdot \min\{d_F(\sigma', \tau[v_1,v_i]) : |\sigma'| \le k\}$. The working storage is $O((k\varepsilon^{-1} + \varepsilon^{-(\alpha+1)})\log\frac{1}{\varepsilon})$. Each vertex in the stream is processed in $O(k\varepsilon^{-(\alpha+1)}\log^2\frac{1}{\varepsilon})$ time for $d \in \{2,3\}$ and $O(k\varepsilon^{-(\alpha+1)}\log\frac{1}{\varepsilon})$ time for $d \ge 4$.

---

*This paper will appear at the International Symposium on Computational Geometry (SoCG 2025)

†Department of Computer Science and Engineering, HKUST, Hong Kong. Email: scheng@cse.ust.hk, haoqiang.huang@connect.ust.hk

‡University of Science and Technology of China, Hefei, China. Email: lejiang@mail.ustc.edu.cn