Local Complexity Adaptable Trajectory Partitioning via Minimum Message Length

Charles R. Twardy, Anthony Stefanidis

Center for Geospatial Intelligence
Department of Geography and Geoinformation Science
George Mason University
{ctwardy, astefani}@gmu.edu

ABSTRACT

We present a minimum message length (MML) framework for trajectory partitioning by point selection, and use it to automatically select the tolerance parameter ε for Douglas-Peucker partitioning, adapting to local trajectory complexity. By examining a range of ε for synthetic and real trajectories, it is easy to see that the best ε does vary by trajectory, and that the MML encoding makes sensible choices and is robust against Gaussian noise. We use it to explore the identification of micro-activities within a longer trajectory. This MML metric is comparable to the TRACLUS metric – and shares the constraint of abstracting only by omission of points – but is a true lossless encoding. Such encoding has several theoretical advantages – particularly with very small segments (high frame rates) – but actual performance interacts strongly with the search algorithm. Both differ from unconstrained piecewise linear approximations, including other MML formulations.

Index Terms— Trajectory, Partitioning, Segmentation, MML, MDL, TRACLUS

1. INTRODUCTION

Line partitioning is important to a variety of applications ranging from road extraction in satellite imagery to blood vessel extraction in medical imagery [1]. Considering trajectory partitioning in particular, we can identify diverse approaches like the use of piecewise arc fitting algorithms to smooth vehicle trajectories [2], the use of dynamic programming for the segmentation of a trajectory into piece-wise polynomial parts [3], or even the use a velocity and orientation extrema to segment sign language hand trajectories [4].

In this paper we present our approach to trajectory partitioning through a novel extension of the Douglas-Peucker algorithm that incorporates an analysis of the local complexity of trajectory segments. We use minimum message length (MML) metrics [5] to express local trajectory complexity and the corresponding goodness of fit of its partitioning, and analyze this information in order to (i) automate the trajectory partitioning process, allowing it to adapt to the complexity of the trajectory itself, bypassing the need for a user-provided information, and (ii) identify highly localized micro-activities within a longer trajectory. For the scope of this paper we consider 'activities' to be equivalent to trajectory patterns.

Regarding micro-activities in particular, the basic premise is that individual trajectories comprise a low frequency activity (e.g. moving from one side of the field of view to the other at a relatively steady pace) and micro-activities superimposed on them (e.g. deviating momentarily from this path because in order to contact another person or object). The approach we present in this paper allows us to detect such micro-activities by analyzing trajectory data.

The above-mentioned extensions make our approach suitable for video-based surveillance applications, an application we intend to pursue in future work, along with direct empirical comparisons to previous metrics.

This paper is organized in the following manner. In Section 2 we develop our approach, and in Section 3 we describe our data and results, followed by concluding remarks in Section 4.

2. TRAJECTORY PARTITIONING USING MML

For a trajectory P defined as a sequence of points $P = p_1 p_2 \ldots p_n$, its partitioning is defined as an approximation of P by a subset of its own points. Accordingly, the partition of P is a sequence of points $Q = q_1 q_2 \ldots q_m$ such that $m \leq n$ and each q_j corresponds to some $p_i, j \leq i$. For discussion purposes, let "segment" refer to adjacent points in P, and let "component" refer to adjacent points in Q. The distinction is artificial.

The line partitioning technique that is of interest for this paper is the Douglas-Peucker (DP) algorithm. Since its introduction [6] the DP algorithm has become a classic reference approach for line partitioning, and still remains the subject of various extensions and improvements [7], [8]. The DP algorithm approximates a trajectory $p_1 \ldots p_n$ by recursively splitting the component $p_1 p_n$ at the point p_i farthest from that line, until no interior points are farther than a pre-selected tolerance parameter ε (expressed for example as a number of pixels).

This well-known algorithm is fast and useful in practice, but has a major shortcoming: the tolerance parameter ε is arbitrary, and there is no metric by which to compare different choices of ε, other than visual inspection. Although we do not treat it here, MML metrics also provide a principled way to handle the multidimensional case when other dimensions are not spatial.

2.1. Message length based partitioning

A key objective of good partitioning is to remove extraneous detail while preserving key features of the trajectory, balancing parsimony and precision, also known as simplicity and goodness of fit. This notion can be formalized using data compression: select the model that leads to the best overall compression – the shortest message length. A model with more parameters can fit better, but those parameters must be stated. At one end is the null model, with a raw transmission of the trajectory (e.g. in pixel coordinates). At the other end is a complex model that matches all the points precisely, even the noise. The shortest message (best model) is usually found in between.

The model or hypothesis *H* is the specific set of components, and $L(H)$ is the length in bits of the description of *H*; the data *D* are the actual points, and $L(D|H)$ is the length of the data encoded with the hypothesis. We seek to minimize the total message length $L(H) + L(D|H)$. The partitioning scheme in TRACLUS [9] is a representative example.

In TRACLUS, the parsimony term, $L(H)$, is the sum of the logs of the Euclidean length of the segments, and the precision term $L(D|H)$ is the sum of the distances between each component and the original trajectory segments it encloses. Distance is a combination of perpendicular and angular distance. We can improve in a couple of ways. The chosen $L(H)$ is zero for components of length 1, and negative for components of length < 1, so trajectories with high frame rates would not get simplified.

Secondly, $L(D|H)$ encodes only distances, so the original data is not recoverable. It even specifies $L(D|H) = 0$ when the component coincides with an original point, which conveys information for free. In sum, the encoding favors extra components, often requiring ad-hoc adjustments to increase partition length [9].

2.2. An MML encoding for DP partitions

The goal is to represent a given trajectory polyline *P* by a subset *Q*, where *Q* simplifies *P* by omitting points that add little information. For reference, the "raw" encoding assumes that the *n* points in *P* are distributed uniformly within the image plane $[0..x_{max}, 0..y_{max}]$, so for a point *p* it takes $L(p)$ bits. For a 640 by 480 image, $L(p)$ =18.23 bits per point. As with TRACLUS, the receiver is assumed to know the total number of points *n*.

However, the message must state *m*, the number of points in *Q*; the q_j are stated using a raw encoding, yielding message length $mL(p)$. Then the omitted or interior points can be encoded more succinctly using their distance parallel and perpendicular to the component: $(D_{||}, D_{\perp})$. Each component $q_j q_{j+1}$ should lie close to its interior points, so we select a coding scheme for $D_{\perp}$ that uses fewer bits for points close to the line. We expect samples to be uniform along the component, so $D_{||}$ is uniform in Euclidean length of the partition. Both real numbers must use a quantum δ.

The encoding first states *m*, the number of components in *Q*, using a uniform prior U (1..*n*-1), where *n*-1 is the number of segments in *P*. Then it is natural to encode each component separately. Let $\mathbf{q} = Q_j$ denote a single component of Euclidean length |**q**|, and let *L*(*p*) denote the message length for raw encoding of a point *p*. The expression for $L(Q_i)$ has these parts:

- State the start and end points: $2 \times L(p)$.
- State the number of interior points as U(1..*r*), hence $\log_2 r$, where *r* is the number of remaining interior points not accounted for in previous segments. [1]
- For each point, state its coordinates as distance parallel and perpendicular to the approximating component: $L(D_{||}) = \log_2(|\mathbf{q}| + 1)$, and $L(D_{\perp})$ is described below.

Subtracting $(m-1)L(p)$ for shared endpoints, the total message length is then $L(m) + \sum_j L(Q_j) - (m-1)L(p)$. It remains to define $L(D_{\perp})$ and mention special cases.

Assume that $D_{\perp}$ follows a Gaussian distribution centered on the component and of unknown standard deviation σ. The MML encoding of Gaussian distributions is well-established, and formulas can be found in, e.g., [5]. One must state σ; then for each point, $L(D_{\perp}) = L(N(0, \sigma))$. We used the PyMML software [10], deriving a subclass for known (stated) mean. We adopt a uniform prior on σ with minimum equal to the quantum, δ = 0.5, and maximum $x_{max}/10$. There are two cases where the prior is violated: 1) $\sigma = 0$ because all $D_{\perp}$ are the same; in this case $D_{\perp}$ need be stated only once; 2) otherwise, forcing a raw encoding. We used a hyperprior of 0.5 (1 bit) for the usual case, and 0.25 each (2 bits) for the "σ=0" and "otherwise" cases.[2]

In addition, a complex trajectory may have points whose projections fall outside **q**, rendering the chosen $D_{||}$ encoding invalid. In this case we use raw encoding, which requires that all components state whether they use raw or regular encoding. (If $|\mathbf{q}| \leq 2$, we can deduce raw encoding.)

[1] Some finesse is possible. If there are *t* total points remaining in the trajectory and *s* remaining segments including this one, then $r = t - 2s$. In the final segment, *r* can be deduced. However, this introduces dependencies.

[2] These are kept bland enough that progressive searches can drop a single point. We also considered the log* universal prior on the Integers (see [11]), but the Gaussian encoding was almost always more efficient.

This encodes both the polygonal approximation, and the goodness of fit, where fit is the sum of $L(D_\perp)$, the cost of stating the residuals given the component and σ. Variations in fit can be used to identify micro-activities, for example by clustering.

2.3. Relation to previous MML partitioning work

Oliver, Baxter, and Wallace [12] considered piecewise linear regressions of univariate time-series, where the *x* values are regular and known. Allison, Fitzgibbon & Dowe [13] extended and improved this. The approach we develop here is adapted to DP-style partitions where the cut-points belong to *P,* and so is suited for automatic selection of ε in the many applications of DP. In addition, we consider identification of micro-activities by examining residuals at several scales (values of ε). Future work could merge features of this paper with [13].

3. TESTING

In order to demonstrate the performance of our approach we used both synthetic and real trajectory datasets. We generated five synthetic trajectories to reflect various classes of activities: a periodic one, a periodic corrupted by noise, a periodic with a localized linear (spike) section, a trajectory that comprises two different patterns, and a random one. Trajectory dataset 'sine' is a sine wave of amplitude 80 pixels; 'sine_dev' replaces a middle section with linear deviations; 'sine_circ' replaces a middle section with loops; 'gauss' is pure Gaussian noise, and 'sine_gauss' is a noisy sine wave. Using standard DP partitioning, at ε = 115, these synthetic datasets are all approximated by a single line. We also have a number of trajectories captured with OpenCV [14] from a fixed camera observing pedestrians. We discuss results for the synthetic trajectories and then for the trajectories extracted from video.

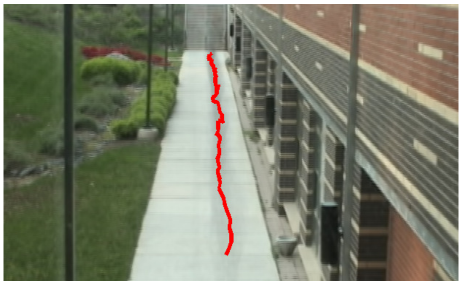
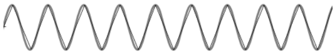
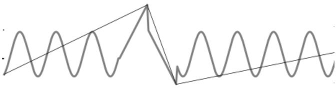
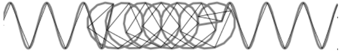
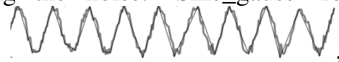


Figure 1: Sample pedestrian trajectory.

3.1. Synthetic Trajectories

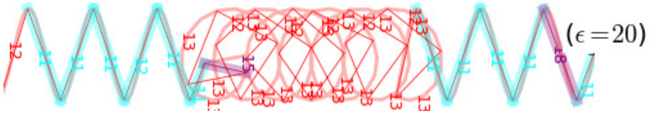
Figure 2 shows message lengths for Douglas-Peucker partitions ('DP') at various ε. Both 'sine' and 'sine_dev' reward small epsilon while the others penalize overfitting. For 'gauss', ε < 6, is worse than raw encoding. Consider 'sine' versus 'sine_dev'. Since 'sine' has amplitude 80, message length does not change until ε < 80. By ε = 35, the DP approximation is a sawtooth wave  and remains stable until ε < 5 when it starts to fit the curves. In contrast, the ε = 90 refinement to 'sine_dev' worsens the fit, incurring a large cost for sending the residuals: . Subsequently, message length decreases until ε < 2. The local deviations in 'dev' are linear, and so encode quite well.

In contrast, 'sine_circ' reaches a minimum at ε = 20, having separated the loops from the sines, but not smoothed the loops, . The loops are not well-modeled by a Gaussian, and additional refinements lose more to model cost than they gain in data cost.

Both 'sine_gauss' and 'gauss' *are* well-modeled, so the model resists fitting the noise. 'Sine_gauss' reaches a minimum at ε = 10, , a slight refinement of the sawtooth pattern first found at ε = 50. Finally, 'gauss' is best modeled by a single component, although it has a local minimum at ε = 25. For ε < 6, it costs more than sending the raw data.

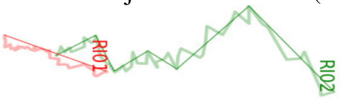
The plateaus in Figure 2 reveal stable features across scales: the DP approximation remains unchanged for several ε. The original 'sine' has a plateau once the sawtooth captures the basic oscillation. 'Sine_circ' also has a sawtooth plateau, near the global minimum, and a second when the loops are finally captured. 'Gauss' has no strong plateaus. These cross-scale plateaus correspond to distinct activities (i.e. trajectory patterns). A new plateau at a finer ε indicates a micro-activity overlaid on an activity identified at a coarser ε.

The residuals in each segment can be tracked and segments clustered by residuals. One such measure is bits per point (bpp). The following shows the result of running *k*-means clustering on bpp with *k* = 3, for 'sine_circ'. The circular micro-activity clusters with relatively high bpp compared to most of the sawtooth.



3.2. Video-captured trajectories

Figure 3 plots message length versus ε for Douglas-Peucker for the eight captured trajectories; once again the best value for ε varied, showing the value of locally adaptable DP.

We see trajectories RI01 (red) and RI02 (green) here, , with the best DP for each. The other fits are similar.

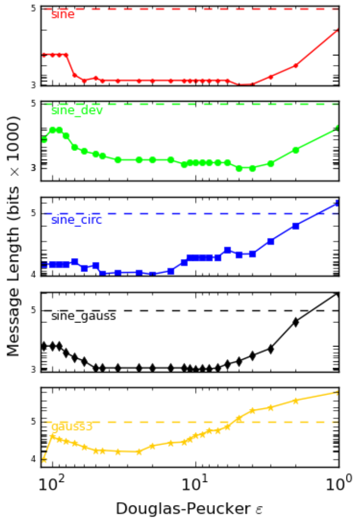


Figure 2: Message lengths for the synthetic trajectories. Lower is better. The dashed line is for raw encoding, always the same; *y* scales vary slightly, as shown.

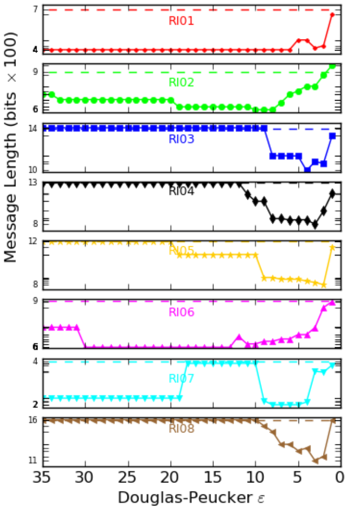


Figure 3: Message lengths for 8 captured trajectories. Lower is better. The dashed line is for raw encoding, which varies by trajectory, as do the scales.

4. CONCLUSION

We presented a minimum message length (MML) framework for trajectory partitioning by point selection, and used it to automatically select the ε parameter for Douglas-Peucker partitioning, offering the potential for adaptation to local trajectory complexity. We also demonstrated multiscale and clustering analyses to identify micro-activities within a longer trajectory.

5. ACKNOWLEDGMENTS

This work was supported by the National Geospatial-Intelligence Agency through a NURI grant, and by Draper Lab.

REFERENCES

[1] Jian Chen, Y. Sato, and S. Tamura, "Orientation space filtering for multiple orientation line segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 5, pp. 417-429, 2000.

[2] P. Banerjee and R. Nevatia, "Dynamics Based Trajectory Segmentation for UAV videos," in *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 345-352, 2010.

[3] R. Mann, A. Jepson, and T. El-Maraghi, "Trajectory segmentation using dynamic programming," in *Object recognition supported by user interaction for service robots*, pp. 331-334.

[4] W. Kong and S. Ranganath, "Automatic hand trajectory segmentation and phoneme transcription for sign language," in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, pp. 1-6, 2008.

[5] C. S. Wallace, *Statistical and Inductive Inference by Minimum Message Length*. Springer-Verlag, 2005.

[6] D. H. Douglas and T. 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*, vol. 10, no. 2, pp. 112-122, Oct. 1973.

[7] L. Li and W. Jiang, "An improved Douglas-Peucker algorithm for fast curve approximation," in *2010 3rd International Congress on Image and Signal Processing*, pp. 1797-1802, 2010.

[8] J. Ma, S. Xu, Y. Pu, and G. Chen, "A real-time parallel implementation of Douglas-Peucker polyline simplification algorithm on shared memory multi-core processor computers," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, pp. V4-647-V4-652, 2010.

[9] J. Lee, J. Han, and K. Whang, "Trajectory clustering: a parition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*, p. 593, 2007.

[10] P. F. Harrison, *PyMML*. 2005.

[11] L. Allison, "Integer Distributions," *Minimum Message Length, MML*, 24-Sep-2003. [Online]. Available: http://www.csse.monash.edu.au/~lloyd/tildeMML/Discrete/Integers/.

[12] J. Oliver, R. Baxter, and C. Wallace, "Minimum message length segmentation," in *Research and Development in Knowledge Discovery and Data Mining*, vol. 1394, Springer Berlin / Heidelberg, 1998, pp. 222-233.

[13] L. Fitzgibbon, L. Allison, and D. Dowe, "Minimum Message Length Grouping of Ordered Data," in *Algorithmic Learning Theory*, vol. 1968, Springer Berlin / Heidelberg, 2000, pp. 56-70.

[14] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.