A Structure-Aware Method for Crowd-sourced Sparse-to-Dense GPS Trajectory Image Generation

Ying Zhang[1*], Yifang Yin[2], Roger Zimermann[3], Zhiwen Yu[1]
[1]*Northwestern Polytechnical University, Xi'an, China*,
[2]*Institute for Infocomm Research (I²R), A*STAR, Singapore*,
[3]*National University of Singapore, Singapore*
[*]Corresponding author: izhangying@nwpu.edu.cn

***Abstract*—Crowdsourced vehicle trajectories are valuable resources to interpret city road networks in a fresh and live manner. Recently, trajectories have been popularly used together with other image data to facilitate various downstream location-based services, among which the very first step is to convert the trajectories data to an image layer. To achieve satisfying results, all these works require densely sampled GPS trajectories which is however sometimes unavailable. So this study explores the possibility to generate dense trajectory images from sparse trajectory data. In order to address the unique features of the trajectory images, we introduce a spatial structure-aware model with a novel triplet-based training strategy and a global-local loss. Experimental results on real-world trajectories show that by using just 1% of trajectory data we can restore most of the original information.**

***Index Terms*—Location based service, GPS, crowdsource, sparse data**

I. Introduction

With the popularity of ride-hailing services such as Uber in the U.S. and DiDi in China, an increasing number of crowd-sourced vehicle GPS trajectories are being collected [9] and become as a powerful resource to infer city road networks. Compared to commercial maps, they reflect the latest real-world road details in a live and fresh manner and it can be flexibly customized for diverse semantics.

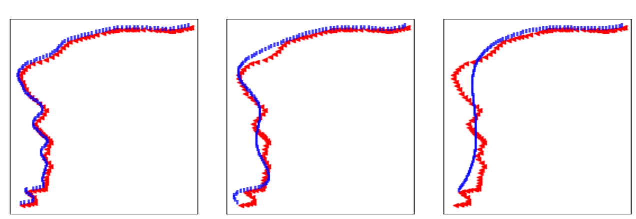
Recently, an increasing number of visual tasks indicate that the integrating of trajectories improves the performance of the original visual task. For example, aerial images are used with trajectories for better road detection [6], lane detection [2] and map image translation [8] where the trajectory data is first rendered into an image layer and then fused with image data for downstream tasks. However, the performance highly depends on the trajectory sampling rate and a sparse trajectory usually leads to a negative impact. Low-sampling rate GPS trajectories have been popularly discussed in the geospatial research community as a sequence of discrete points but rarely in a visual form. This is mainly because that (1) Mapping GPS points to an image pixel is a many-to-one process which further reduces the GPS sampling rate. (2) The problem is exacerbated if (other) image data is captured at a coarse level, *i.e.*, each pixel covers a larger region and thus discretizes the trajectory further.

This work was supported by the National Natural Science Foundation of China (No. 61960206008, 62272390, 62025205, 62072375, 62032020) and also by Singapore Ministry of Education Academic Research Fund Tier 1 under MOE's official grant number T1 251RES2029.

To generate such a dense layer, majority methods works on independent trajectory processing methods, a potential way is through interpolation, the quality of which however degrades severely when lowering the GPS sampling rate (Fig. 1). The second category of methods leverages external road knowledge to recover the information, *e.g.*, map matching assigns a GPS point to a road and uses that road's center-line as a dense trajectory but such knowledge might be unavailable [5], [7]. Thus, this study investigates a new problem, namely *crowd-sourced sparse-to-dense GPS trajectory image generation*, which aims to render a visually well-connected image (a dense layer) for a given geospatial space from a collection of only low-sampling rate trajectories. We expect this dense layer to reflect the road topology similarly to using high-sampling rate trajectories.

The key contributions of this paper are:

- To the best of our knowledge, this is the first time the sparse-to-dense trajectory generation problem is investigated in visual form. The dense trajectory images can benefit many downstream location-based tasks that work on visual data.
- We propose a structure-aware triplet training strategy and loss to address the unique features of the trajectory images. Using only 1% of GPS data we can mostly recover the road topology at a quality similar to using 100% of the original GPS data.
- Experiments on real-world data indicates our method performs clearly better than a few classic methods.



(a) 100 p/s (b) 50 p/s (c) 1 p/s

Fig. 1: The trajectory interpolation quality degrades with lower GPS sampling rates (red/blue: original/interpolated).

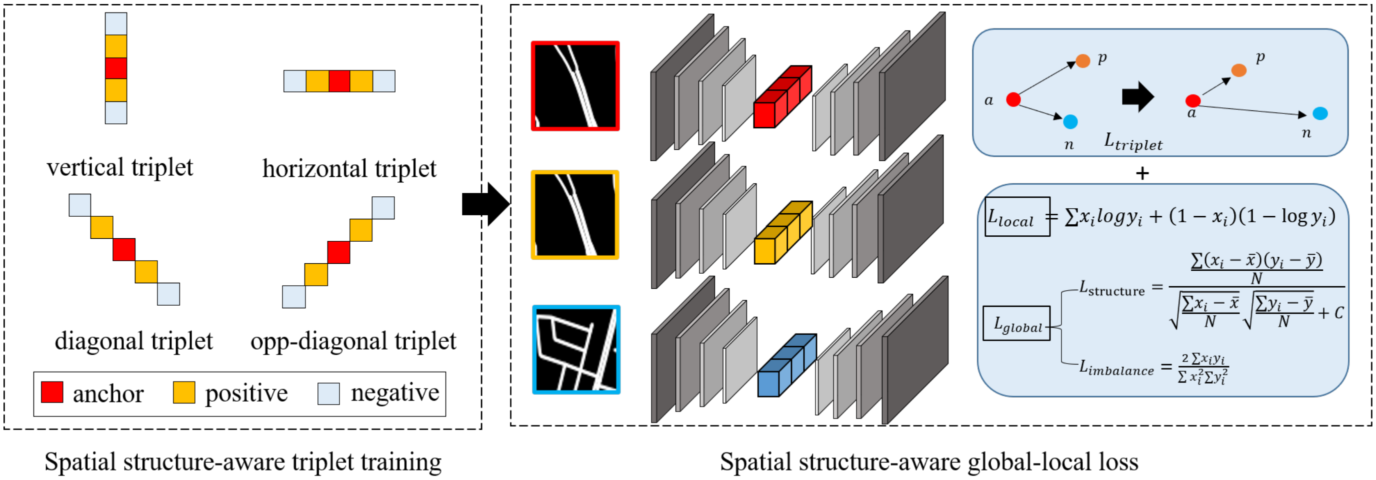


Fig. 2: The overview of the proposed model whose input is the sparse trajectory image and output is the dense trajectory image. The model is featured with (a) a spatial structure-aware triplet training, which considers the direction-specific data to be composed as the training input, and (b) a spatial structure-aware global-local loss, which addressed the highly-structurized characteristics of the trajectory images.

II. Method

A. Problem formulation

Denote a GPS point as l which is a pair of longitude and latitude, a GPS trajectory is of T continuous GPS points and could be represented by $J = [l_1 \to l_2 \to \dots l_T]$. Given a collection of GPS trajectories $\mathcal{J} = \{J_1, \dots, J_k\}$, we define **a GPS trajectory image** I by rendering a two dimensional array with $N = W \times H$ pixels from $\mathcal{J}$. Each pixel covers a geospatial area whose span is determined by the downstream applications, which is usually parameterized by the World Geodetic System as a positive integer from a given range. If the time interval between two continuous GPS points is below a threshold, we name them as high-sampling-rate trajectories and the corresponding trajectory image is named as a **sparse trajectory image**. Otherwise we will have the high-sampling rate trajectories and the corresponding **dense trajectory image**. Under this setting, our problem is to convert a sparse trajectory image to a dense version. For each pixel, we are going to predict its label as an intensity level and this paper considers a binary level for simplicity.

We adopt an encoder-decoder structure for pixel-wise prediction and use U-Net as the backbone model. However, unlike the natural images whose contents are mostly smoothed distributed over the whole space, the valid contents of the trajectory images' (pixels with GPS records) only exist in a few parts and the distributions follow clear spatial topology. Therefore, a conventional training might not be sufficient and we design a novel spatial structure-aware model as in Fig. 2. We will explain its two featured modules – a spatial structure-aware triplet training and a spatial-structure global-local loss design in below.

B. Spatial structure-aware triplet training

Roads are continuously spanning across the geospatial space in various directions. Consider that the nearby trajectories are more likely to cover the same roads or their segments which have similar geometrical shape, so the spatially close-by trajectory images should also share similar visual contents. Thus, we introduce a new training strategy to leverage such spatial relations for visual feature learning. Specifically, we prepare the training data as triplets $\langle a, p, n\rangle$, where a is an anchor, p is its adjacent trajectory images (positives) and $\langle n\rangle$ are non-adjacent trajectory images (negatives). As above-mentioned, nearby trajectory images are more similar than the far-away ones, there is therefore a distance relation among the three elements in a triplet as shown in the equation below:

$$D(e(a), e(p)) < D(e(a), e(n)) \tag{1}$$

where $D(A, B)$ is a distance measurement between A and B, and $e(A)$ is a function to extract the pattern from the input trajectory image A. In our case, $e(A)$ is the flattened version of the bottleneck feature from the encoder. Next, we define an anchor's adjacency as its neighborhood regions. Under the grid-cell setting, let $C_{(x,y)}$ denote the cell at a location (x, y), and denote a radius parameter as an positive integer r, the adjacency of cell $C_{x,y}$ is a set of nearby cells: $A = \{C_{(x-i,y-i)} \cup C_{(x+i,y+i)} \cup C_{(x,y-i)} \cup C_{(x,y+i)} \cup C_{(x-i,y)} \cup C_{(x+i,y)} \cup C_{(x+i,y-i)} \cup C_{(x-i,y+i)} | i = 1 \sim r\}$. We see that adjacency covers the nearby cells from all directions. However, a road in real world is usually extendable only to certain direction and therefore we introduce the direction-specific adjacency which only covers the cells located along one of the following directions: horizontal, vertical, diagonal and opposite-diagonal ones. Based on this, we have four types of direction-specific triplets. As shown in the left part of Fig. 2, the positives and negatives of an anchor will only be selected from the same direction. Therefore, the relation in Eqn. 1 is extended to the following direction-specific version:

$$D(e(a), e(p)) < D(e(a), e(n)|\Re(n) = \Re(p)) \quad (2)$$

where $\Re(p)$ and $\Re(n)$ are the triplet type of the positives and the negatives, respectively. To move towards reflecting such a relation, we use the contrastive loss to regulate the training:

$$\begin{aligned} L_{triplet}(a, p, n) &= max(0, D(e(a), e(p)) + m \\ &\quad -D(e(a), e(n))) \end{aligned}$$

where m is a value to ensure a minimal separation between two pairs. The above loss is used in the training to further maintain the spatial-correlation.

C. Spatial structure-aware global-local loss

Our problem is a pixel-wise binary classification and we require the last decoder layer to be softmax activated with a larger value meaning a larger probability. In order to obtain a pixel-wise binary prediction, denote x and y as the prediction image and the ground truth image (dense trajectory image), respectively, we adopt the cross-entropy loss to guide the training:

$$L_{local}(x, y) = \sum_{i=1}^{N}(x_i \log y_i + (1 - x_i)(1 - \log y_i)) \quad (3)$$

where N is the total number of pixels in an image. This loss emphasized the local correctness, we can regard it as a local loss. As mentioned above, trajectory images contain very structuralized visual contents which is reflected at a relatively global manner to show the underlying road topology (e.g., horizontal, vertical lines and curves). Therefore, we also integrate a global loss to address their structure differences:

$$L_{stucture}(x, y) = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})/N}{\sqrt{\sum(x - \bar{x})/N} \times \sqrt{\sum(y - \bar{y})/N} + C} \quad (4)$$

where the denominator is composed of the standard variances of the predicted and ground truth dense image with $\bar{k}$ is an average of k. C is a small constant weight to avoid a division by zero. Numerator is the co-variance of the two images. On the other hand, since the valid pixels of a trajectory image only occupies small image percentage [6], it is necessary to address the class imbalance and we use a dice loss:

$$L_{imbalance}(x, y) = \frac{2\sum_{i=1}^{N} x_i \cdot y_i}{\sum_{i=1}^{N} x_i^2 + \sum_{i=1}^{N} y_i^2} \quad (5)$$

Based on these deductions, the overall loss finalized in Eq. 6 is used for the dense trajectory image learning.

$$L = \alpha_1 L_{local} + \alpha_2 L_{structure} + \alpha_3 L_{imbalance} + \alpha_4 L_{triplet} \quad (6)$$

where α is a weighted factor.

III. Experiments

Real-World GPS Dataset We evaluated our method on the publicly available real-world vehicle trajectories dataset over two cities [3] with key data statistics summarised in Table I. By default, we prepare the GPS image coordinates according to WGS84 system. For the dense trajectory images, the size of a region covered by an image depends on the zoom level specified for the tile. The default zoom level is 16, at which the streets are identifiable. We render the trajectory images by checking if there exists at least one GPS point in each pixel-region. For sparse trajectory images, we randomly selected 1% of the original GPS data to generate the images. If there are no GPS points in a region, the sparse images and their corresponding dense GPS images are discarded.

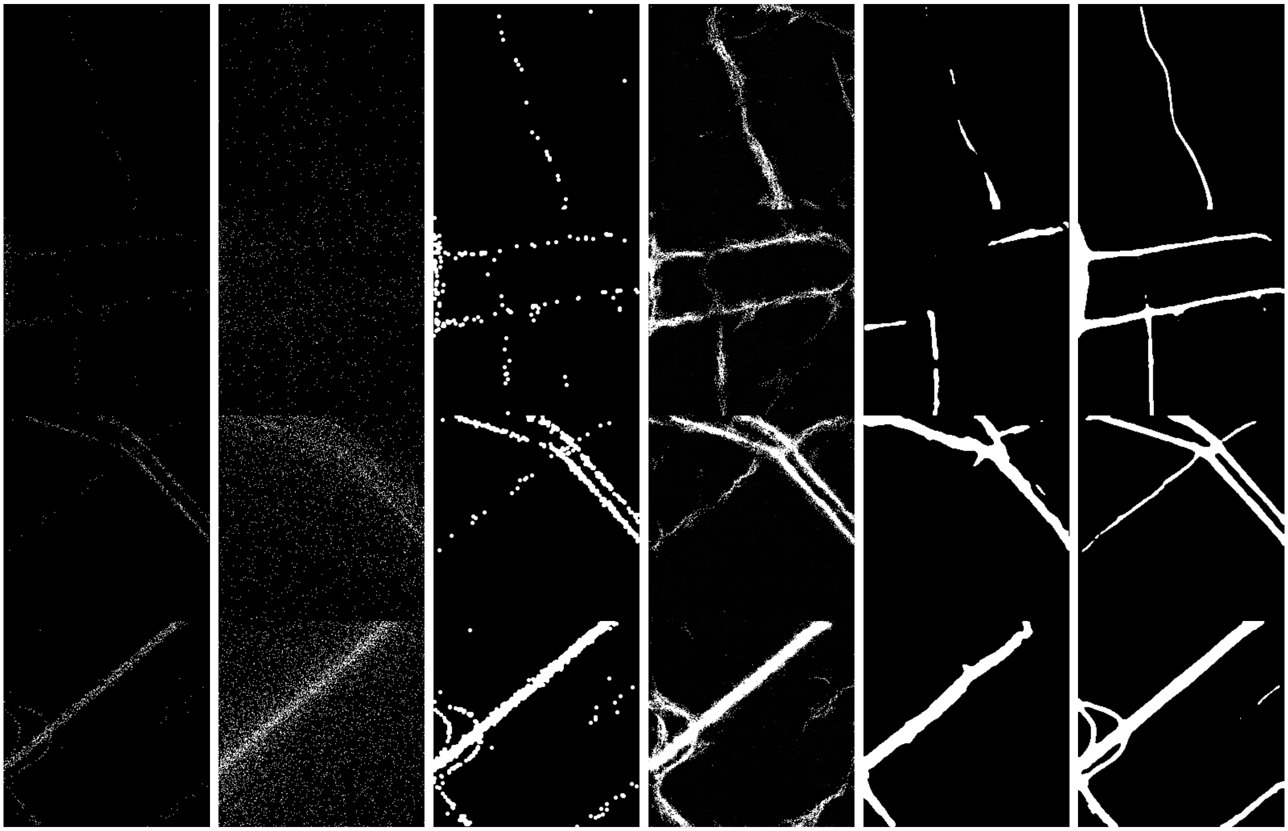
TABLE I: The data Statistics of two experimental cities.

Attributes	City 1	City 2
Area (km^2)	24.86 x 46.28	227.19 x 106.04
#GPS Points	30,329,685	55,988,420
Original GPS Rate	~1/sec	~1/sec
#Trajectories	28,000	55,995
Default Zoom Level	16	16
Image Resolution	256×256	256×256

Baselines We evaluate a few potential methods for this new problem: *Spline Interpolation* up-samples the sparse GPS records by a factor of 100× with spline interpolation, *Kernel Density Estimation (KDE) [1]* is a non-parametric method to estimate the data distribution and we threshold it as a binary version for the estimated dense GPS records, *Pix2Pix [4]* is an adversarial model to translate a pair of images between two domains and we carry out the sparse to dense trajectory images translation, *Sat2Dense* replaces Pix2Pix's input to a satellite image. Among these methods, Spline Interpolation and Gaussian KDE process each trajectory individually while the remaining methods process a group of trajectories at together.

TABLE II: Evaluation results and ablation study.

City 1			
Model	mIoU	F1	Accuracy
Spline Interpolation	0.5226	0.2222	0.9211
Gaussain KDE	0.7105	0.6305	0.9619
Pix2Pix	0.7058	0.6300	0.9537
Sat2Dense	0.6913	0.6107	0.9454
Proposed, w/o triplet	**0.7861**	**0.7533**	**0.9695**
Proposed	**0.7909**	**0.7602**	**0.9700**
City 2			
Model	mIoU	F1	Accuracy
Spline Interpolation	0.5195	0.1633	0.9503
Gaussain KDE	0.6688	0.5398	0.9686
Pix2Pix	0.7000	0.6046	0.9676
Sat2Dense	0.6960	0.5964	0.9679
Proposed, w/o triplet	**0.7773**	**0.7330**	**0.9770**
Proposed	**0.7879**	**0.7490**	**0.9779**



(a) Input (b) Interpolation (c) KDE (d) Pix2Pix (e) Sat2Dense (f) Proposed

Fig. 3: Comparison results. The four examples (top to bottom) have increasing GPS occupancy percentages in images. Both Interpolation and KDE are based on individual trajectory processing while the remaining methods jointly convert a group of trajectories into one image.

Results As shown in the Table II, our method has a clear improvement on both cities over baselines cross multiple metrics. The results of City 1 are slightly better than City 2, which might due to the existence of a larger amount of small and short routes in the City 2's road network and this may introduce some additional difficulties. We also visualize some example images in Figure 3 and our method shows an overall improvement and could for example generate straighter roads with better connectivity.

Ablation Study One feature of our method is the spatial-location aware triplet training and we show the results if without this component in the last two rows in Table II. The performance drops from 0.05-1.6% on both cities which indicates the effectiveness of this regulation.

IV. CONCLUSIONS

This paper presents a new problem statement about how to render a dense image from sparse GPS trajectories. The generated dense trajectory images can benefit many location-based tasks that use CNNs originally designed for visual information only. Our approach potentially relieves the burden to collect dense trajectory data and is suitable for many practical scenarios that are faced with constraints for sense trajectory collection. The proposed spatial structure-aware method is effective on real-world datasets, where using just 1% of trajectory data can we restore most of the original information.

REFERENCES

[1] J. Biagioni and J. Eriksson. Map inference in the face of noise and disparity. In *GIS*, 2012.

[2] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun. Dagmapper: Learning to map by discovering lane topology. In *CVPR*, 2019.

[3] X. Huang, Y. Yin, S. Lim, G. Wang, B. Hu, J. Varadarajan, S. Zheng, A. Bulusu, and R. Zimmermann. Grab-posisi: An extensive real-life gps trajectory dataset in southeast asia. In *ACM SIGSPATIAL International Workshop on Prediction of Human Mobility*, 2019.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017.

[5] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 352–361, 2009.

[6] T. Sun, Z. Di, P. Che, C. Liu, and Y. Wang. Leveraging crowdsourced gps data for road extraction from aerial imagery. In *CVPR*, 2019.

[7] Y. Yin, R. R. Shah, G. Wang, and R. Zimmermann. Feature-based map matching for low-sampling-rate gps trajectories. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 4(2):1–24, 2018.

[8] Y. Zhang, Y. Yin, R. Zimmermann, G. Wang, J. Varadarajan, and S.-K. Ng. An enhanced gan model for automatic satellite-to-map image conversion. *IEEE Access*, 8:176704–176716, 2020.

[9] J. Zheng, W. Sun, S. Huang, S. Shen, C. Yu, J. Zhu, B. Liu, and H. X. Liu. Traffic signal optimization using crowdsourced vehicle trajectory data. Technical report, 2018.