Datenbank-Spektrum
https://doi.org/10.1007/s13222-024-00484-3

SCHWERPUNKTBEITRAG

Multi-Modal Contextualization of Trajectory Data for Advanced Analysis

Paul Walther[1] · Fabian Deuser[1,2] · Martin Werner[1]

Received: 31 May 2024 / Accepted: 20 August 2024

Summary
Rising amounts of generated geospatial data, either trajectory-like tracking data, raster-like imagery, or vector-like mappings as in OpenStreetMap (OSM), grow the need for multi-modal algorithmic analysis. Existing machine-learning-based algorithms contradictly mainly focus on image and textual input representations and cannot deal with other modes of geospatial data. Therefore, we propose a novel method to contextualize vector-like trajectory data with surrounding data to create easy-to-be-analyzed image-like representations. Our approach includes the proposition of a chase-cam-like scanline over space according to the trajectory's speed and possibly smoothed orientation. Thereby, surrounding pixels in the vicinity of the trajectory points are accumulated along the scanline and are combined into a visual representation of the trajectory. To show the potential effects of our work, we predict traffic regulations for trajectory sections in the vehicle speed dataset based on our proposed trajectory-based sampling of orthophotos in the same region. This proposes a new way of using multi-modal data sources (trajectories and airborne imagery) to extract road metadata.

Keywords Trajectory analysis · Trajectory contextualization · Geospatial raster image · Scanline

1 Introduction

Increasing amounts of mobile devices and tracking sensors generate more and more moving point objects, respectively, trajectory data [24]. GPS-based localization in outdoor environments, and indoor localization systems in warehouses or private environments, e.g., based on WiFi, result in incidental collection of this data. Additional trajectory data is created while using social media and web pages, as well as while taking phone calls [27]. Researchers in recent years focused on optimizing the analytical approaches to investigate this data type [24]. They proposed various methods ranging from visualization tools [30, 10] to clustering and pattern discovery [24, 21], classification [4, 7] and prediction methods [19, 2]. So far, most available algorithms use trajectories as a single and only type of information, though this limits analysis possibilities.

At the same time, increased governmental and societal interest led to a growing amount of satellite and airborne imagery in the form of raster datasets, e.g. the European Copernicus satellite program[1] and the INSPIRE program of the European commission[2]. This raster data is often stored in compressed patches, i.e., the data is subdivided into distinct spatial areas and stored in a compressed format within this context.

Additionally, mapping missions generate more high-detailed vector representations like the prominent example of the OpenStreetMap (OSM)[3]. The advantage of these representations is that they are generally well-structured and are available in various levels of detail (LOD).

Therefore, using the manifold geospatial data sources described before holds enormous potential in improved meth-

✉ Paul Walther
paul.walther@tum.de

Fabian Deuser
fabian.deuser@unibw.de

Martin Werner
martin.werner@tum.de

1 TUM School of Engineering and Design, Department of Aerospace and Geodesy, Technical University of Munich, Lise-Meitner-Straße 9, 85521 Ottobrunn, Bavaria, Germany

2 Institute for Distributed Intelligent Systems, University of the Bundeswehr Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Bavaria, Germany

[1] https://www.copernicus.eu/, last access: 27.09.2024.
[2] https://inspire.ec.europa.eu/, last access: 27.09.2024.
[3] https://www.openstreetmap.org/, last access: 27.09.2024.

Published online: 07 October 2024

ods for fields like mapping, agriculture, traffic analysis, and catastrophe management. For example, in remote regions, today's mapping is already often facilitated by remote sensing, e.g., in the community-run HOT tasking manager of OSM[4]. Adding additional data sources like tracking data, e.g. by vehicles, to these approaches may help automate previous manual efforts and thereby increase the precision and timeliness of the created maps. Furthermore, the manifold nature of data capture allows for map validation and thereby can be used for fighting vandalism in community-run tools like OSM [18].

In essence, leveraging more data sources will consistently enhance data-based decision-making. However, this presents a challenge: How can we incorporate the different data types into one solution considering the Big Data nature of geospatial data? For this problem, we propose a new approach to analyze trajectory data together with its context with the following key properties:

- We propose a new multi-modal representation of trajectory data based on sampling surrounding environment features from aerial imagery or vector maps.
- Trajectories are visualized in images, where columns of the image are sampled from raster images surrounding the trajectory. This allows for visual interpretation of trajectory data by both machine learning models and humans.
- We show that our approach can predict street features like surfaces or speed limits from the proposed novel representation of trajectories.

2 Related Work

A *basic trajectory* T is defined as a sequence of m points p, such that $T = [p_1, p_2, ..., p_m]$, with $p = \{x_1, ..., x_n, t\}$. The coordinates x_i denote the position in $\mathbb{R}^n$, and t is the timestamp at which the position was captured [7, 4]. So far trajectories are analyzed in various ways: With *trajectory classification* the main goal is to assign a label to the trajectory data or subsets of the trajectory [4]. If there are no previous labels available various *clustering* methods (dense, hierarchical, spectral) can be used to still group trajectories [4]. Both techniques can be useful in various applied fields, for example, object motion prediction, traffic monitoring, activity understanding, outlier detection, weather forecasting, and geography [4]. Other analysis fields for trajectories are the *similarity prediction of trajectories* [21, 24, 11] and *privacy preserving analysis* [14].

In comparison to trajectories, *geospatial raster data* is stored in a grid-like structure with equal-sized spacing between raster points [22]. Geospatial raster data is mostly stored and provided in patch-like partly overlapping representations of single raster images, called mosaics. These come with different levels of detail. Raster data is generally not made or stored to effectively access or modify single values or pixels [1], as the patches can be compressed and must be loaded into memory to be decoded. Therefore, storing raster images in databases as binary blobs is discouraged [1]. This raises the question of efficiently accessing individual elements of raster data [25]. In the context of our problem, this boils down to efficiently finding raster points related to a vector-based trajectory. To solve this issue, various approaches were presented in the literature on jointly analyzing vector and raster data [25]. [8] uses a scanline to calculate zonal statistics of vector-denoted areas in raster datasets. In addition, to enable compressed storage with improved accessibility, research is being conducted on raster data structures [13]. For example, k^2-trees, as an improved version of a region quadtree, showed improved range queries in the raster domain while still lacking performance in single cell queries [25, 5]. Alternatively, approaches for in-memory querying were proposed by [3].

The first approaches of *contextualizing trajectories*, which means enriching the pure trajectory data with additional information, were performed by adding semantic information, also called aspects, to the location passed by the trajectory [29, 21]. Contexts are thereby denoted as "influential factors [...] during the move" [24] and can be differentiated into external contexts (e.g., geographical and environmental factors) and internal contexts (e.g., states, circumstances, and properties like speed) [17]. In general, many authors emphasize the value of this additional information for trajectory analysis [24, 29, 6]. Though, [24] and [11] also explain, that only limited research has been conducted on the influence of contexts of trajectories on analyzed features like similarity. Examples of trajectory contextualization are, e.g., the use of weather information as data cubes around trajectories to predict aircraft routes, resulting in improved trajectory predictions [2]. Another approach showed that introducing context priors may improve trajectory parsing in video images [15]. Also, for clustering trajectories [29] and movement pattern mining [6], the contextualization was described to be beneficial. Based on previous work, so far trajectories have been contextualized with various additional data, though, to the best of our knowledge, this contextualization has not been constructed from the spatial surroundings of the observed trajectories.

3 Methods

In comparison to previous approaches that add aspects as additional information to the single trajectory points, we

[4] https://tasks.hotosm.org/, last access: 27.09. 2024.

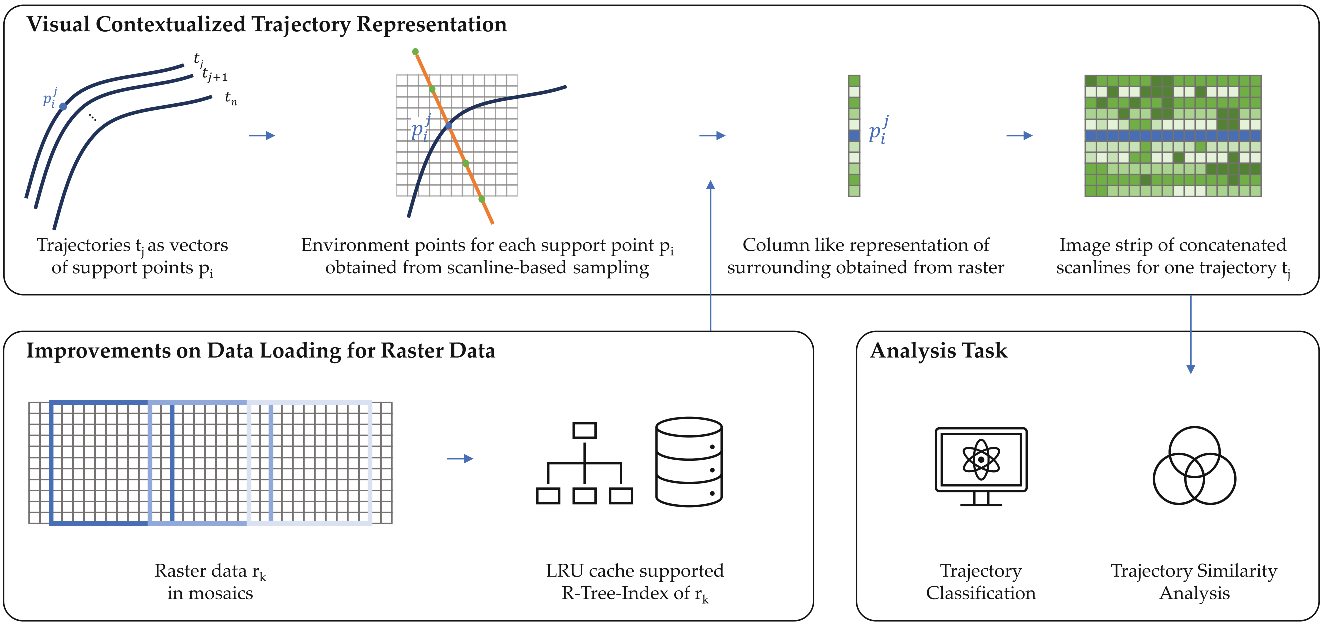

Fig. 1 Our scanline-based raster sampling approach contextualizes trajectories in representations of their surroundings

propose a new approach that not only adds additional aspects but also translates the trajectory together with its spatial context in an image-like representation. In the following, we explain our approach based on the three questions:

- How to represent contextualized trajectories as images?
- How to efficiently load single image points?
- How to use this approach for image-enhanced trajectory classifications?

Figure 1 gives an overview of the proposed method.

3.1 Visual Contextualized Trajectory Representation

Suppose we want to include multi-dimensional information in general. In that case, there are two possibilities for appending information: We can embed the environment features in context aspects and add those to the trajectory data. Alternatively, we can embed the trajectory in a more complete format that additionally stores information on the locality of the data points in a data cube around trajectory points. This may include storing additional information by introducing additional local coordinate frames at every point in the trajectory to store multi-dimensional information around a trajectory point. This approach is envisioned in Fig. 2. In general, there are several ways to orient this local coordinate system. While the orientation of the y-axis is given in 2D trajectories on the earth's surface by the geospatial nature of the data defining an orthogonal on the map, a rotation around this axis is generally possible for different analyses of trajectories. The z-axis can be tangential to the trajectory (as shown in Fig. 2), in the direction of the next trajectory point or the direction of the finish point of the whole trajectory or a trajectory segment. The problem with this approach is that no well-developed algorithms for analyzing these data cubes exist.

Consequently, we propose to use analysis algorithms originally proposed for other multi-dimensional data representations, namely images, to analyze trajectory data. This requires reformating the trajectory data and the respective data cubes around them into an image-like representation. Our approach thereby focuses on 2D trajectories in the xz-plane. Considering that we want to apply additional knowledge from areas around the trajectory anyway, we propose embedding each trajectory point into a column of image points sampled from a raster representation in the xz-plane. The column-like sample of points is conducted on a *scan-*

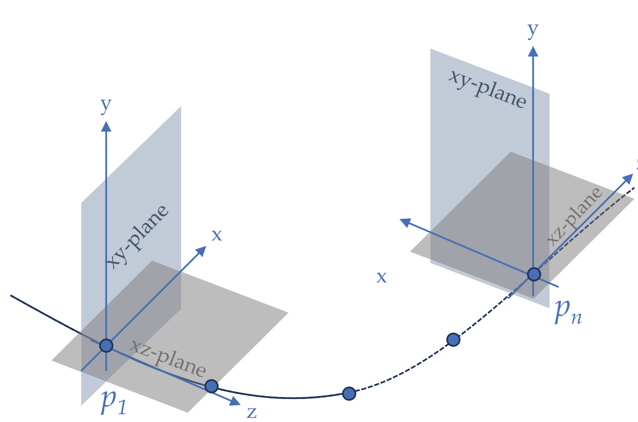

Fig. 2 Introduction of local coordinate frames along the trajectory

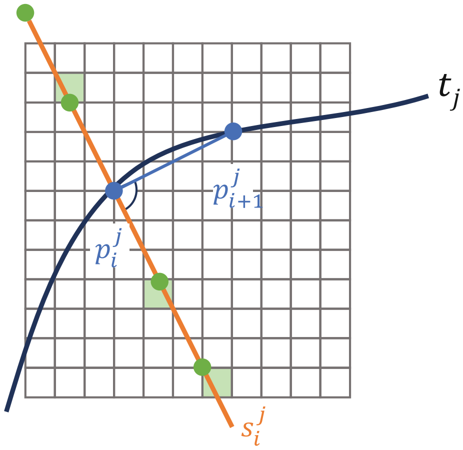

Fig. 3 Visualization of the sampling approach, points on the scanline s_i^j denote the determined location of the sampling point; the marked cells show the sampled raster cell

line in the xz-plane going through a trajectory point, later called support p_i. This idea is shown in Fig. 3. In comparison to the scanline-based sampling of raster points described by [8], our scanline approach is fundamentally different. Indeed, it is not used to sample subsets of a raster dataset denoted by a polygon in the vector space but instead extends the vector space of the trajectory by an additional dimension.

Returning to the nature of trajectories as a space-time path, a first question arises on selecting support points for the image representation. This allows various methods also depending on the nature of the storage of the trajectory. Generally, a selection of these points on the trajectory may happen in an equal distance, equal time, data creation-dependent, or parameter-dependent manner. Parameters can thereby be additional aspects of the trajectory and its context. As in the practical geospatial domain, trajectories are often captured by storing GPS-based location signals. In this case, failures in measurement can lead to an introduction of jitter in the trajectory. Therefore, smooth curvatures can not be guaranteed for the trajectories. The outlier effect of single distorted trajectory points can be reduced by smoothing the trajectory with additional sampled points.

A second question arises about the location and orientation of the scanline used for sampling image points in the raster domain: Generally, every differentiable curve in the xz-plane may be considered a valid approach for this scanline and the construction of the best curve for an analysis task may be use case dependent. All straight/linear lines or function-based trajectories are considered valid scanlines. Another parameter is the angle of the scanline to the coordinate system. For our proposal and the simplicity of our approach, we only consider straight scanlines in the direction of the x-axis of the local coordinate frame. As a baseline approach, we defined this local x-axis as orthogonal to the straight connection of the support point p_i with the following support point p_{i+1} as shown in Fig. 3.

The third parameter in this scanline approach is the actual sampling of points on this scanline. Again, these may be equally distributed in a certain space interval around the support point or sampled according to defined distribution (random, Gaussian, etc.). For our approach, we use equal space sampling as a baseline method.

3.2 Improvements on Data Loading for Raster Data

The nature of the raster image storage is a main issue in conducting the proposed sampling of raster points to obtain the visual images from trajectory data and corresponding 2D images. As they are stored in compressed mosaic patches r_k (compare Fig. 1) and cannot be directly accessed pixelwise, the standard approach expects a loading of the whole image patch into memory to retrieve single pixels. Garbage collection would then ensure the cleaning of this memory space as soon as the image is not actively used anymore, which would result in several reloads of r_k to get the visual representation of single trajectories. On a global scale, processing trajectories in similar spatial regions must load the same image patch.

Therefore, in our approach, we propose a threefold procedure. First, we use an R-tree index to fast identify the correct image patch for a given location. Second, we order all trajectories to be contextualized spatially based on their bounding boxes, such that proximate trajectories are contextualized in sequence. Finally, we apply a least-recently-used (LRU) cache to memory to keep the least loaded mosaics r_k opened and untouched by garbage collection. Based on the nature of trajectories, which implies the spatial proximity of consecutive raster points, and additionally, the spatial sorting of the trajectories to be contextualized one after another, this reduces the amount of reloads of a single image mosaic patch drastically. This is due to the fact, that there is a high probability that drawn samples for the following support point will lay on a similar image patch as for the previous one. With our approach we therefore keep as many images as possible in active RAM, thereby improving the RAM utilization and avoiding unnecessary and expensive reloading of images. With that, only a fraction of image-loading procedures are necessary compared to the standard approach. To load the single raster pixel, after the location of the to-be-sampled point is obtained from the scanline approach and the loading of r_k in RAM, a distance-based selection is chosen. More complex selection of raster values like the Bresenham Algorithm [9] are not applicable as they are too inefficient.

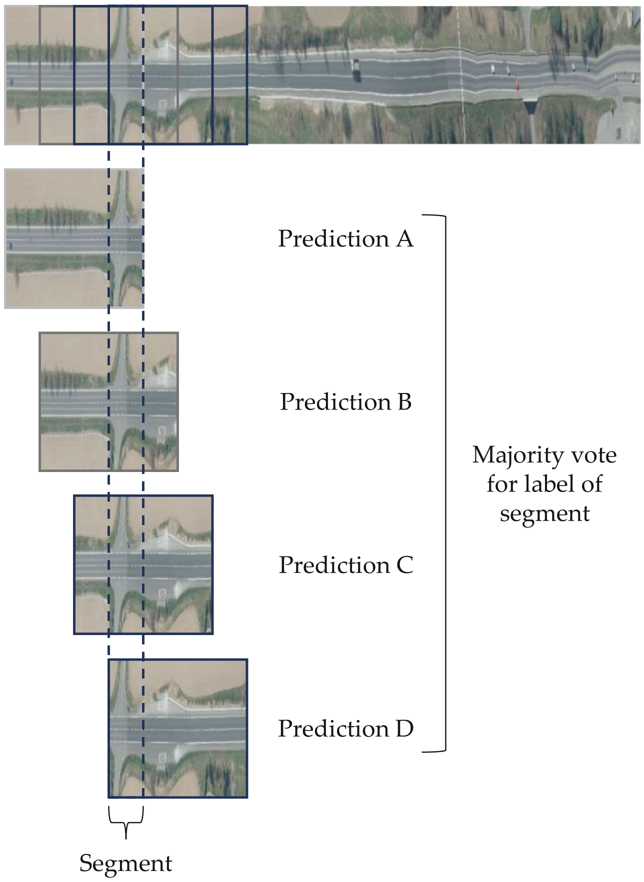

Fig. 4 Patching of trajectory image stripes and majority voting for segment classification

3.3 Analysis Task: Image Classification

Our proposed sampling approach results in an image strip of the size *# trajectory support points* p_i *×# sampling points per trajectory point*. While the second parameter can be fixed during creation for a certain set of trajectories, the first parameter is purely dependent on the length of each trajectory. Common image-based analysis methods like Convolutional Neural Networks (CNNs) expect images within one dataset to be of constant size. Therefore, we propose tiling the images produced with our approach into equally sized patches. As explained before, trajectory analysis often involves classifying trajectory points and segments. Our approach can do so by classifying each scanline separately, which means support point by support point or continuously classifying image segments of predefined length (amount of pixels). While in the ideal case, an instance-based classification of the support points would result in the most accurate classification, the missing context of single points can lead to misconceptions and wrongly inhomogeneous classifications of actually homogeneous trajectory segments. Therefore, our approach features a segment-based majority vote by providing multiple labels for each road segment. Therefore we use a segment-based majority voting algorithm as shown in Fig. 4 to classify road segments.

4 Datasets

For our approach, we relied on two types of data: Trajectories in vector representation (in our case the vehicle speed dataset (VSD)[26]) and additional context information in raster data format (here orthophotos of the Czech government [12]).

The *VSD* [26] contains more than 9000 km trajectories tracked as GPS tracks on unfamiliar routes in the Czech Republic. The dataset contains 5973 individual rides, which are augmented with labels from OSM, like maximum allowed speeds or road classification. A detailed explanation of used classification labels in our approach is given in Table 1.

As depicted in the plot shown in Fig. 7, the data exhibits a skewed class distribution. This skewness is attributed to the intrinsic characteristics of road types within the Czech Republic. Considering the example of the predominance of asphalt roads along with other factors, such as varying speed limits, it is to be expected that the dataset would exhibit a bias towards certain characteristics. Consequently, this uneven distribution underscores the complexity of tasks related to road classification.

The Czech Republic's Land Survey Office, in cooperation with the Military Geographic and Hydrometeorological Office, periodically presents *orthophotos of the whole Czech territory* [12]. For this paper, data spanning the whole territory totaling 30,228 patches with around 10,000 × 15,000 pixels each summing up to a total of more than 4E12 pixels was downloaded. The disk size of the jpeg2000 compressed data is ≈1500 GB. A pixel in this data is normalized to a spatial extent of 12.5 × 12.5 cm. Figure 5 gives an example of a used orthophoto.

To *prepare the data* for subsequent analysis tasks, a split into three sets, training, validation, and test, is conducted

Tab. 1 Label overview obtained from the VSD [26]. All labels are originally from OSM.

Road Feature	Explanation
way_maxspeed (Speed Limit)	*'10', '15', '20', '30', '40', '50', '60', '70', '80', '90', '100', '110', '130'*
way_type (Category)	*'residential', 'tertiary', 'motorway_link', 'trunk_link', 'secondary_link', 'secondary', 'tertiary_link', 'service', 'primary_link', 'trunk', 'unclassified', 'motorway', 'primary', 'living_street'*
way_surface (Category)	*'gravel', 'compacted', 'sett', 'cobblestone', 'asphalt', 'paving_stones', 'unpaved', 'concrete', 'paved'*

Fig. 5 An example of an orthophoto (Z1_1M_N49d29m00s_E017d16m00s) obtained from [12]

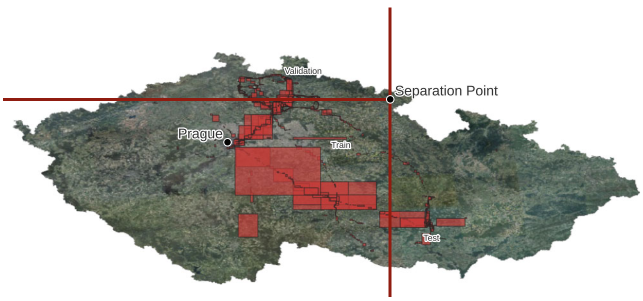

Fig. 6 Separation of the data for our training splits. We show the locality of the trajectories in the Czech Republic, with trajectories visualized by their bounding box and background orthophoto obtained from [12]

based on the geolocation of bounding boxes of the trajectories depicted in Fig. 6.

This avoids information leakage between them as there is no risk of predicting results on an image that was seen before in the training process. The separation point is chosen at the coordinates N50.600 E16.170. The three resulting sets have an approximately equal class distribution as visualized in Fig. 7.

5 Experiment

To prove the validity of our approach, we conducted an experiment to predict road features based on trajectories and their visual surroundings. The experiment is performed in two steps: First, trajectory segments are contextualized in visual representations before a classification algorithm is trained to predict the road features.

5.1 Trajectory Contextualization

Since the trajectories of the VSD are more or less uniformly sampled in time, as described by the authors [26], we use all sampled points as support points p_i^j for the respective trajectory t_j For the trajectory embeddings, we uniformly sampled points on scanlines orthogonal to the connection of the support point p_i to the following support point p_{i+1}. The length of the scanline was denoted to be 40 m in the image coordinate reference system, and a total of 256 points were sampled with an equal distance sampling on this line. This resulted in separate image stripes of width 256 pixels and varying lengths l_j for each trajectory t_j. These were tiled in 256 ×256 pixel patches with a stride size s of 16. Trajectories t_j with less than 256 support points and the last $n = \text{count}[p_i^j] \bmod s$ support points for each trajectory were discarded.

This resulted in a total of 476,942 image patches. As explained before, data division for training, validation, and testing follows a regional separation of the trajectories. The size of the three sets of training, validation and test are 294,337, 73,161, and 94,340 patches, respectively.

5.2 Classification Task

To assess the applicability of trajectory-derived data, we employed a CNN trained on 256×256 pixel orthophoto patches. We utilized the ConvNeXt architecture [16] for multi-label classification tasks, specifically targeting *way_type*, *way_surface*, and *way_maxspeed* categories. Each category was independently classified using a ded-

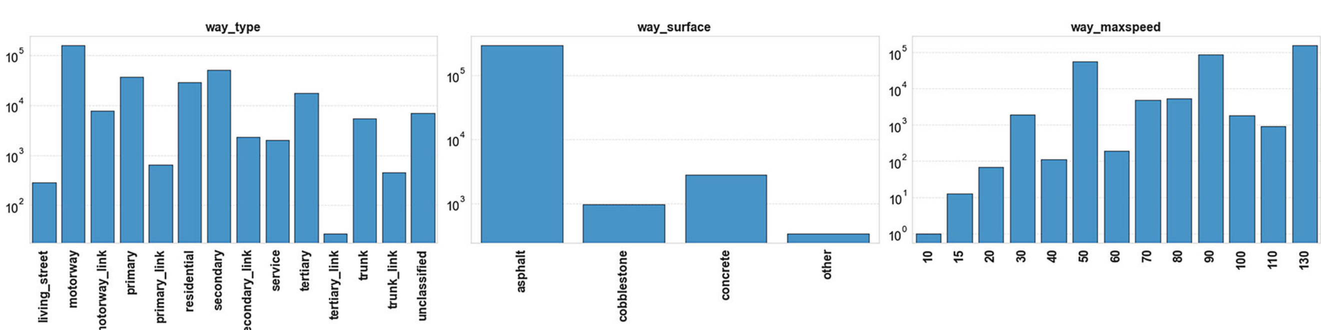

Fig. 7 Visualization of the training splits in logarithmic scale for the number of occurrences for each label. All classes are available in the train split, but we observe not all classes during validation and testing

icated classifier built on CNN-extracted features. Label details for these categories are provided in Table 1.

Implementation Details: We train for 26 epochs the pre-trained ConvNeXt-Base with a learning rate of $1e^{-4}$ and a Cosine Decay schedule with a warm-up of 1 epoch. As a loss, we use the Asymmetric Loss for Multi-Label Classification as proposed by [23]. This loss is especially suited for skewed data distributions in multi-label settings. Additionally, we augment the images with random rotation, horizontal flipping, color jitter, blurring, sharpening, grid dropout, and coarse dropout.

6 Evaluation and Results

With the proposed experiment, we achieve good results in giving a visual interpretability of trajectory data as shown in Fig. 8. Not only the environment of a trajectory can be visually examined with this representation, but also the distortion in commonly recognizable image parts of a defined shape and size like cars, trees, or crossings allows additional statements about the velocity traveled (when support points are sampled on an equal time approach). Issues with our proposed data generation approach especially occur in more densely populated areas. There, two factors result in distortions of the image: On the one hand, the multiple sharp turns result in the overlap of an adjacent scanline and, therefore, unnecessarily distort the image by repeating patterns, as visualized in Fig. 9. On the other hand, the additional distortion of GPS signals is resulting in non-smooth trajectories.

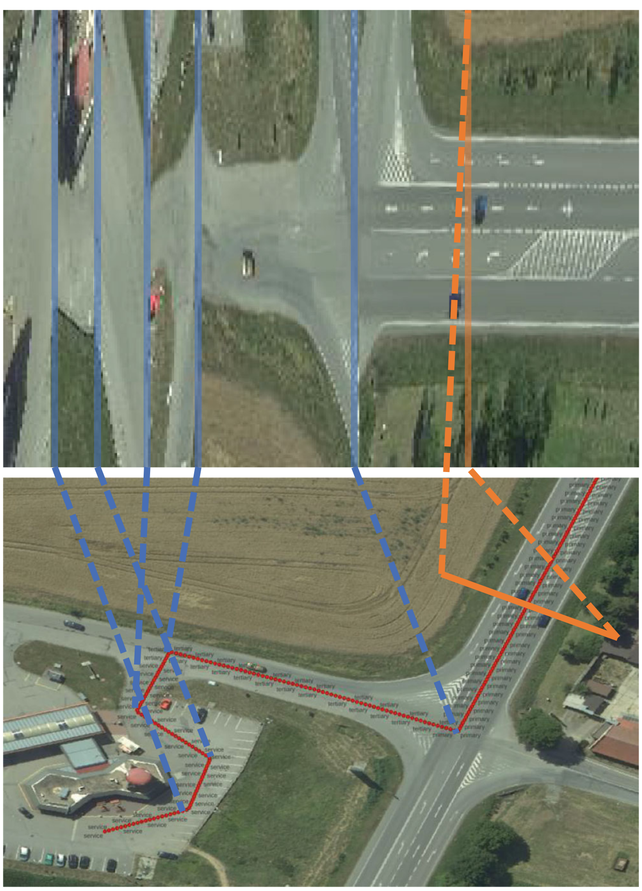

Fig. 8 Orthophoto with plotted trajectory points on the street and corresponding visual embedding generated with our proposed approach. Dark markers denote turns, and the light line gives an example of a scanline and the respective visualization in the embedding

For an evaluation of the prediction of road features, classification results are presented in Table 2. The ZeroR classifier, a baseline model, predicts solely based on class distribution ("stratified" approach), essentially rendering its predictions as random, guided by class frequencies [20]. In contrast, our CNN model exploits the intrinsic image features, outperforming the ZeroR baseline. This delineates the effectiveness of CNNs in extracting and utilizing complex patterns within images for classification, affirming the model's capability to surpass mere probabilistic guessing, as demonstrated by the comparative results.

While the classifications of way types and speed limits show promising results, it is obvious that the strong skewness of labels for the way surface prediction tasks (compare Fig. 7) causes difficulties for the classifier, resulting in worse results. However, this also expresses the complexity of the posed classification tasks itself.

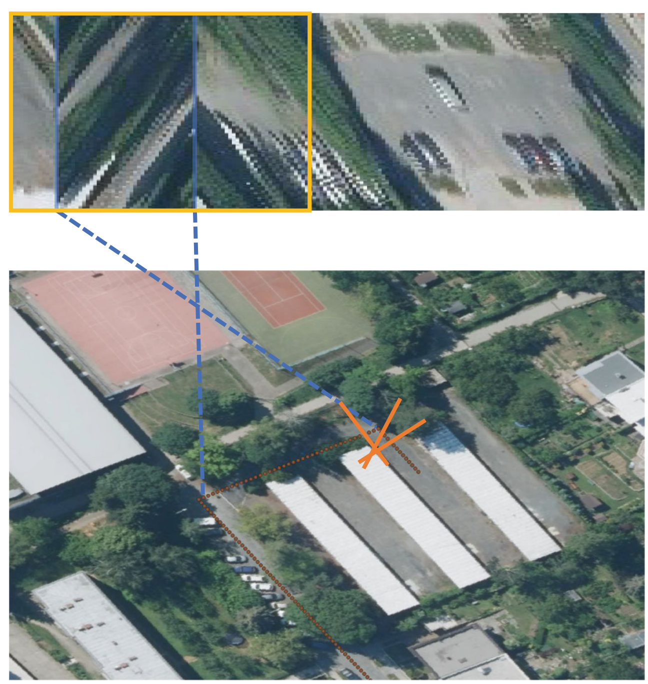

Fig. 9 Possible distortion due to sharp trajectory turns; points single raster areas are sampled multiple times (compare light rectangle in the top and crossing scanlines in the bottom)

Tab. 2 Classification results of our CNN approach on multiple tasks compared to the ZeroR random classifier

Approach	Way Type		Way Surface		Speed Limit	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Validation						
ZeroR	0.03	0.08	0.24	0.25	0.08	0.09
ConvNeXt	0.28	0.18	0.38	0.25	0.23	0.20
Test						
ZeroR	0.07	0.07	0.25	0.25	0.08	0.08
ConvNeXt	0.27	0.30	0.25	0.25	0.28	0.25

7 Conclusion and Outlook

With our chase-cam-based selection of trajectory-surrounding raster pixels, we propose a novel way of dealing with multi-modal input data for trajectory analysis.

The visual evaluation showed the representative and interpretable structure of these embeddings. Technical feasibility of our approach as a basis for downstream trajectory analysis by classifying trajectory parts was given with the results obtained in road feature classification based on the Czech vehicle speed dataset [26] and orthophotos of this region [12]. While the general approach worked well, we still propose the following points for future development:

- Reducing artifacts and improving on the exactness of the representation by improving on raster point selection, removing outliers, and rising curve smoothness.
- Improve the database access patterns for single raster elements during sampling by applying more advanced data structures, including randomized data structures as proposed by [28] to improve access performance.
- Adapt the approach to additional data types, such as vector maps, and extend it to the 3D case where we would not have a scanline but a 2D scanplane.

With this work, we improve the general analyzability of trajectories so that more advanced analyses, like automatic semantic descriptions, become possible. This improves data usability and utilization and can help to expand the remote sensing domain, which has so far been based mainly on image data, to include the additional modality of vector data.

Acknowledgements This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 507196470.

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Arc GI (2024) How raster data is stored and managed: ArcMap 10.8 | Help archive. https://desktop.arcgis.com/en/arcmap/latest/manage-data/raster-and-images/how-raster-data-is-stored-and-managed.htm
2. Ayhan S, Samet H (2016) Aircraft Trajectory Prediction Made Easy with Predictive Analytics. In: Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowl. Discov. and Data Min. ACM, New York, NY, USA, KDD '16, pp 21–30. https://doi.org/10.1145/2939672.2939694
3. de Bernardo G, Álvarez-García S, Brisaboa NR et al (2013) Compact Querieable Representations of Raster Data. In: Kurland O, Lewenstein M, Porat E (eds) String Processing and Information Retrieval. Springer Int. Publ., Cham, pp 96–108. https://doi.org/10.1007/978-3-319-02432-5_14
4. Bian J, Tian D, Tang Y et al (2019) rajectory Data Classification: A Review. Acm Trans Intell Syst Technol 10(4):1–34. https://doi.org/10.1145/3330138
5. Brisaboa NR, de Bernardo G, Konow R et al (2016) Aggregated 2D range queries on clustered points. Inf Syst 60:34–49. https://doi.org/10.1016/j.is.2016.03.004
6. Choi DW, Pei J, Heinis T (2017) Efficient mining of regional movement patterns in semantic trajectories. Proc Vldb Endow 10(13):2073–2084. https://doi.org/10.14778/3151106.3151111
7. Leite Da Silva C, May Petry L, Bogorny V (2019) A Survey and Comparison of Trajectory Classification Methods. In: 8th Brazilian Conf. on Intell. Syst. (BRACIS), pp 788–793. https://doi.org/10.1109/BRACIS.2019.00141
8. Eldawy A, Niu L, Haynes D et al (2017) Large Scale Analytics of Vector+Raster Big Spat. Data. In: Proc. of the 25th ACM SIGSPATIAL Int. Conf. on Adv. in Geogr. Inf. Syst. ACM, New York, NY, USA, SIGSPATIAL '17. https://doi.org/10.1145/3139958.3140042
9. Gaol FL (2013) Bresenham Algorithm: Implementation and Analysis in Raster Shape. J Comput 8(1):69–78. https://doi.org/10.4304/jcp.8.1.69-78
10. Guo H, Wang Z, Yu B et al (2011) TripVista: Triple Perspective Visual Trajectory Analytics and its application on microscopic traffic data at a road intersection. In: 2011 IEEE Pacific Vis. Symp, pp 163–170. https://doi.org/10.1109/PACIFICVIS.2011.5742386
11. Hamedi H, Shad R (2022) Context-aware similarity measurement of lane-changing trajectories. Expert Syst With Appl 209:118289. https://doi.org/10.1016/j.eswa.2022.118289
12. Koukalová B (2022) Archive Orthophoto of the Czech Republic. https://cuzk.cz/CZ-00025712-CUZK_ORTOARCHIV

13. Ladra S, Paramá JR, Silva-Coira F (2017) Scalable and queryable compressed storage structure for raster data. Inf Syst 72:179–204. https://doi.org/10.1016/j.is.2017.10.007
14. Li S, Shen H, Sang Y (2020) A survey of privacy-preserving techniques on trajectory data. In: Shen H, Sang Y (eds) Parallel Architectures, Algorithms and Programming. Springer Singapore, Singapore, pp 461–476. https://doi.org/10.1007/978-981-15-2767-8_41
15. Liu X, Lin L, Jin H (2013) Contextualized Trajectory Parsing with Spatio-Temporal Graph. IEEE Trans on Pattern Anal and Mach Intell, p 1. https://doi.org/10.1109/TPAMI.2013.84
16. Liu Z, Mao H, Wu CY et al (2022) A ConvNet for the 2020s. In: 2022 IEEE/CVF Conf. on Comput. Vis. and Pattern Recognit. (CVPR), pp 11966–11976. https://doi.org/10.1109/CVPR52688.2022.01167
17. Nathan R, Getz WM, Revilla E et al (2008) A movement ecology paradigm for unifying organismal movement research. Proc Natl Acad Sci Usa 105(49):19052–19059. https://doi.org/10.1073/pnas.0800375105
18. Neis P, Goetz M, Zipf A (2012) Towards automatic vandalism detection in openstreetmap. ISPRS Int J Geo-inf 1(3):315–332. https://doi.org/10.3390/ijgi1030315
19. Pang Y, Zhao X, Yan H et al (2021) Data-driven trajectory prediction with weather uncertainties: A Bayesian deep learning approach. Transp Res Part C Emerg Technol 130:103326. https://doi.org/10.1016/j.trc.2021.103326
20. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: Machine Learning in Python. J Mach Learn Res 12:2825–2830
21. Petry LM, Ferrero CA, Alvares LO et al (2019) Towards semantic–aware multiple–aspect trajectory similarity measuring. Trans Gis 23(5):960–975. https://doi.org/10.1111/tgis.12542
22. Pingel T (2018) The Raster Data Model. Geogr Inf Sci Technol Body Knowl. https://doi.org/10.22224/gistbok/2018.3.11
23. Ridnik T, Ben-Baruch E, Zamir N et al (2021) Asymmetric loss for multi-label classification. In: Proc. of the IEEE/CVF Int. Conf. on Comput. Vis, pp 82–91. https://doi.org/10.1109/ICCV48922.2021.00015
24. Sharif M, Alesheikh AA, Kaffash Charandabi N (2018) Context-aware pattern discovery for moving object trajectories. Proc ICA 1:1–6. https://doi.org/10.5194/ica-proc-1-102-2018
25. Silva-Coira F, Paramá JR, Ladra S et al (2020) Efficient processing of raster and vector data. PloS one 15(1):e226943. https://doi.org/10.1371/journal.pone.0226943
26. Vrany J, Krepelka M, Srb R et al (2023) Vehicle speed dataset. https://doi.org/10.21227/n1z9-e630
27. Werner M (2015) BACR: Set Similarities with Lower Bounds and Application to Spatial Trajectories. In: Ali M (ed) Proc. of the 23rd SIGSPATIAL Int. Conf. on Adv. in Geogr. Inf. Syst. ACM, pp 1–10. https://doi.org/10.1145/2820783.2820802
28. Werner M (2021) GloBiMapsAI: An AI-Enhanced Probabilistic Data Structure for Global Raster Datasets. ACM Trans Spat Algorithms Syst 7(4):1–24. https://doi.org/10.1145/3453184
29. Ying JJC, Lee WC, Weng TC et al (2011) Semantic trajectory mining for location prediction. In: Proc. of the 19th ACM SIGSPATIAL Int. Conf. on Adv. in Geogr. Inf. Syst. ACM, New York, NY, USA. GIS '11, pp 34–43. https://doi.org/10.1145/2093973.2093980
30. Zhang Y, Klein K, Deussen O et al (2022) Robust visualization of trajectory data. It - Inf Techno 64(4-5):181–191. https://doi.org/10.1515/itit-2022-0036