Hierarchical Prism Trees for Scalable Time Geographic Analysis

Carson J. Q. Farmer[1]* and Carsten Keßler[2]

[1] Geography Department, University of Colorado at Boulder, CO, USA
carson.farmer@colorado.edu
[2] Department of Development and Planning, Aalborg University Copenhagen, Denmark
kessler@plan.aau.dk

Abstract. As location-aware applications and location-based services continue to increase in popularity, data sources describing a range of *dynamic processes* occurring in near real-time over multiple spatial and temporal scales are becoming the norm. At the same time, existing frameworks useful for understanding these dynamic spatio-temporal data, such as time geography, are unable to scale to the high volume, velocity, and variety of these emerging data sources. In this paper, we introduce a computational framework that turns time geography into a scalable analysis tool that can handle large and rapidly changing datasets. The Hierarchical Prism Tree (HPT) is a dynamic data structure for fast queries on spatio-temporal objects based on time geographic principles and theories, which takes advantage of recent advances in moving object databases and computer graphics. We demonstrate the utility of our proposed HPT using two common time geography tasks (finding similar trajectories and mapping potential space-time interactions), taking advantage of open data on space-time vehicle emissions from the EnviroCar platform.

Keywords: time geography, dynamic indexing, spatio-temporal queries, scalability

1 Introduction

Decision making in the corporate, private, and public spheres is increasingly based on spatio-temporal information. These information sources include real-time traffic counts, location-based social-media interactions, environmental sensor networks, as well as space-time trajectories of humans, animals, and vehicles. At the same time, modern advances in information and communication technology have converged with popular culture (*e.g.*, geo-tagging, location-based services, crowd-sourcing, *etc.*) to create an environment that is overflowing with new forms of spatial data [1,2]. Many of these emerging data sources contain details about movements and flows of individuals, objects, or information over

* Corresponding Author

geographic space, and are part of a growing list of dynamic spatio-temporal data sources.

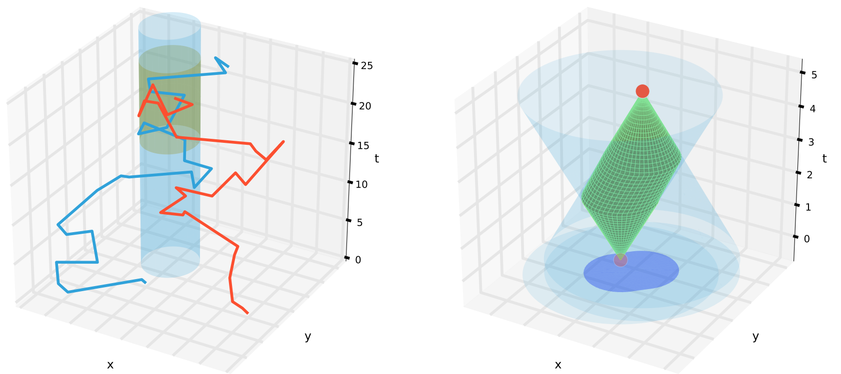
Existing frameworks for understanding dynamic processes *are* available, including the rich conceptual and theoretical frameworks of *time geography* [3,4]. Hägerstrand's time geography was originally developed to understand how human migration activities are constrained at the individual level, and provides an ideal framework within which to explore modern spatio-temporal data sources. Indeed, there has been renewed interest in time geography concepts for geospatial research [5,6], including for location-based services [7,8], accessibility [9,10], trip planning [11,12], and health [13]. Despite this increasing interest, issues of scalability and applicability to emerging data sources are limiting time geography's use in data-intensive research.

While time-geography is useful for thinking about many types of spatio-temporal movements, much of the existing literature focuses on a limited number of individuals or features, and does not generally scale to larger problems. In this paper, we present a computational framework for time geographic analysis that aims to preserve the underpinnings of time geography (and in particular, Miller's [3] time geographic measurement theory), while at the same time increasing the scalability and applicability of the framework to meet the needs of a data-intensive research agenda.

In the following section (Sec. 2), a brief background on time geography is presented, followed by a presentation of dynamic (spatio-temporal) data structures and bounding volume hierarchies (Sec. 3) as a potential means of scaling time geographic concepts. Building on these ideas, a framework (Sec. 4) for the development of time geographic data structures which takes advantage of recent advances in moving object databases and computer graphics research is introduced. Following this (Sec. 5), two examples of this framework applied to common time geographic analysis tasks are presented, using space-time data on vehicle emissions from the EnviroCar platform [14]. We conclude (Sec. 6) with a discussion of the proposed framework and directions for future research.

2 Background

The basic concepts of time geography are the *space-time path*, describing changes in an object's location with time, and the *space-time prism*, describing an object's travel *potential*. This potential is constrained by the speed at which the object can travel (v_{max}), as well as locations at which the object must be present (*e.g.*, home and work when the object in question is a person). In general, a space-time *path* (see Fig. 1a) consists of a sequence of control points and a corresponding sequence of path segments connecting these points. In this definition, control points are observed or measured locations in space and time, and segments connect temporally adjacent control points. A space-time *prism* (see Fig. 1b) may exist between any pair of temporally adjacent control points, creating a time interval during which unrecorded (or future) travel may occur. An object may thus occupy locations in space other than the straight-line segment between two adjacent control points. The outline of the prism represents the limits of the locations that can be visited, as defined by the known space-time control points, and the object's maximum velocity, v_{max}, which defines the prism's diameter.



(a) Space-time paths (b) Space-time prism

Fig. 1. Features of Hägerstrand's time-geography. Concepts include (a) space-time paths, bundling and stations (cylinder), as well as (b) space-time prisms, and accompanying start and end points, future and past cones (semi-transparent), and potential path areas (projected onto base).

The time-geographic concepts above have been formalized by Miller [3], who introduces a rigorous measurement theory based on three key assumptions: (1) the metric space satisfies the notions of identity, non-negativity, and triangular inequality about distance, (2) data are recorded at specific points in time, and (3) analysts have perfect information about the system (although relaxations of this assumption have been explored to some degree [15,3]). Building on these relatively simple assumptions, Miller has developed mathematical (and geometrical) definitions for space-time paths, prisms, stations, bundles (convergence of two or more paths for some shared activity over some given length of time), and intersections (two or more features sharing the same location(s) in space and time). Miller also provides strict conditions within which space-time paths are bundled and where intersections may occur between paths and prisms.

Research areas that typically employ time geography as an analysis tool deal with different aspects of mobility (*e.g.*, location-based services, accessibility, trip planning, health). The proliferation of mobile devices, sensor networks, and new developments such as the Internet of Things create an abundance of new data sources for these domains, which have traditionally dealt with small, easily tractable, and carefully selected samples. In the following sections, we will introduce a computational framework that turns time geography into a scalable analysis tool that can handle large and rapidly changing datasets, allowing the aforementioned domains to leverage these new data sources. We will argue that dynamic spatial indexes are not sufficient in this context, and that dynamically updated bounding volume hierarchies present a viable solution.

3 Dynamic Spatial Indexes

A wide range of data structures have been proposed for efficient queries on spatial and spatio-temporal data [16], including indexing strategies geared towards location-based services [17], real-time data [18], or more general spatio-temporal data [19,20]. However, for objects that may move in space and time, these indexes have to be *continually* updated, which can limit their utility in many cases. In order to address this issue, a number of dynamic indexing algorithms [21,22], including dynamic *spatial* indexes [23,24,25] have been developed, many of which are designed specifically for keeping track of moving objects [26,27,28]. These efforts have lead to a number of useful data structures and indexing schemes for static *and* dynamic spatial data, with a particular focus on 2D geometries (although some innovative exceptions have been proposed [29]). Because time geography embeds objects in 3D *space-time*, it is prudent (and useful) to query and perform analysis on objects in this space directly. For example, while conceptually similar to 2D space plus 1D time, a 3D index allows us to query and explore the *joint* space-time in a more efficient way (rather than querying space and then time or vice versa) and allows us to work directly with 3D volumes, rather than 2D time slices. For this, one can turn to the computer graphics literature, where data models for static and continuously moving 3D objects are required to speed up the rendering process [30,31].

3.1 Bounding Volume Hierarchies

Many 3D spatial indexing (or space partitioning) algorithms, such as R-trees, octrees, and kd-trees, slice 3D space with a flat 3D plane [32] to create sub-volumes. This is efficient to search, but presents a problem when objects overlap the split boundary. In dynamic applications of octrees or kd-trees [21], [33], objects may be placed into all sub-volumes they touch. This requires extra overhead when working with moving objects, and extra tests when traversing the space to handle duplicate occurrences. As such, while kd-trees have excellent performance for *static* geometries [34], when it comes to dynamic settings with multiple moving objects, a different approach is required[3].

Instead of selecting a split-plane to divide volumes, a bounding volume hierarchy (BVH) tree of arbitrary enclosing volumes (*e.g.*, bounding boxes, capsules, cylinders, spheres, *etc.*) can be used [36], [30] (Fig. 3). Here, the sub-volumes of a node don't have a particular split plane dividing them, and instead, the objects are divided to minimize some feature of the sub-volumes (generally the surface-area or volume, estimated by a heuristic). This approach has been shown

[3] Although some parallel versions of kd-trees [35] do show promise.

to display superior construction performance over kd-trees [34], and because objects need not be split across sub-volumes, it also allows for *dynamic* object updates, insertions, and deletions [37], which facilitate dynamic BVH implementations. Furthermore, because the tree contains arbitrary enclosing volumes (*i.e.*, there is no clear split plane), sub-volumes are allowed to overlap. Indeed, the ability for sub-volumes to overlap is one of the main reasons that BVHs can handle efficient dynamic updates. When objects only move a short distance, the only adjustment required is a simple adjustment of the bounds of their enclosing volume(s). Even if the volumes overlap other volumes, the BVH will still function correctly (although at slightly reduced efficiency). Furthermore, the arbitrary enclosing volumes provide a significant level of flexibility, even facilitating nested (or multi-scale) BVHs (*i.e.*, a BVH of BVHs is possible).

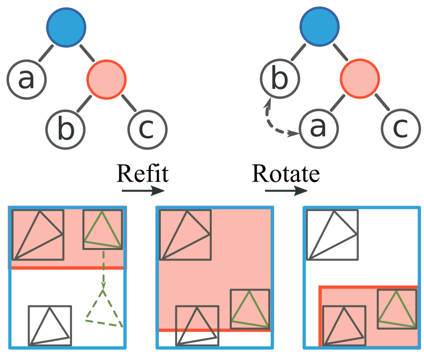


Fig. 2. Tree rotations are local restructuring operations that modify subtrees of a binary tree by swapping direct child and grandchild nodes [38]. In this case, as triangle (c) moves, the bounding volume expands, but rather than splitting the modified node into separate nodes containing triangles (b) and (c), tree rotations allow the BVH to identify and perform helpful merges and splits, such as merging (a) and (c) into a new leaf node.

When there *is* significant overlap, the BVH tree generally needs to be restructured [39]. To perform this re-structuring efficiently, [38] have developed a method based on localized updates to the BVH structure via tree-'rotations' which has proven extremely useful [37], [40]. Combining these tree-rotations with the ability to have overlapping volumes, handling moving objects in a BVH works in two ways: (1) if the movement is minimal, the BVH can be quickly and conservatively expanded to handle the new location (at the cost of efficiency), or (2) if movement is significant (*i.e.*, overlaps are large), tree-rotations to optimize the BVH structure can be performed (Fig. 2).

4 Hierarchical Prism Trees

As mentioned previously, one of the key features of time geography is the space-time prism, a 3D geometric construct that defines *potential* space-time accessibility. Using a BVH tree, prism intersection tests can now be performed on the actual prism volume, rather than at discrete time slices, which is common practice in GIS-based time geography *e.g.* [41,42]. Algorithms for intersection detection of cones and bounding boxes are readily available, many of which are well-tested and efficient[4]. For most time geography analysis, simple bounding box intersection tests provide a quick test of intersection, with more computational tests (cone/cone and cone/cylinder) reserved for intersecting prisms (though in most cases, only a 2D projection of prism intersections is required, not the actual intersection of the prisms).

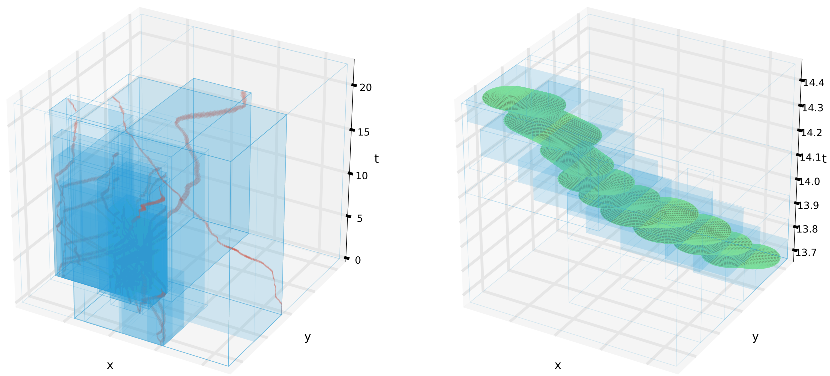
The above BVH techniques can be implemented in a time geographic framework, where the 3D space represents location on the x and y axes, and time on the t (or z) axis (see Fig. 3). The concepts of cones and prisms from time geography mean that *approximate* location queries can be handled using relatively simple collision tests and *predictive* location queries [43] can take advantage of the uncertain nature of cones/prisms. While these types of queries require *a priori* information about an object's behavior (v_{max}), when the prism shape and size are unknown (or likely to be variable), existing methods are available that can be used to estimate features of the space-time prism/cone [42], [44,45]. Furthermore, because the BVH only requires an estimate of the bounding *volume* (*i.e.*, not the geometry of the object itself) to facilitate efficient updates and queries, more fine-grained analyses and queries are able to *lazily* [46] evaluate an object's position and shape, leading to further efficiency gains. Now, space-time intersections, searches, and analyses can be efficiently implemented for a large number of continually moving space-time objects, with minimal computational overhead, within a hierarchical tree of space-time prisms, or a Hierarchical Prism Tree (HPT).

When an even larger number of objects are being tracked, a nested approach to structuring space-time paths and prisms may be required. For instance, rather than tracking each individual space-time prism in a HPT, it may be preferable to track the overall space-time path instead; using the HPT to handle updates of the overall trajectory. When finer-grained details are needed (*i.e.*, to compute joint potential path areas), a nested HPT of space-time prisms can be lazily generated and queried.

5 Examples

In this section, we present two common time geographic analysis tasks which take good advantage of the proposed HPT framework. The first (Sec. 5.2), based on finding similar space-time movement patterns, is somewhat simplistic given the nature of the binary tree solution proposed here. The second (Sec. 5.3), based

[4] See for example, `http://www.realtimerendering.com/intersections.html`



(a) HPT with space-time paths (b) HPT showing space-time prisms

Fig. 3. Dynamic HPT techniques map ideally onto a time geography framework. A HPT of space-time paths (a) can be dynamically built and efficiently queried via time-slice, nearest-neighbor, or bounding box queries, and results can be filtered (b) using more complex intersection tests at the level of space-time prisms (in this case, the HPT in (b) is a subset of the largest space-time path traveling west-to-east in (a)). Tree leafs and nodes are denoted by semi-transparent and empty boxes respectively, with nested trees (a) and space-time prisms (b) as solid objects.

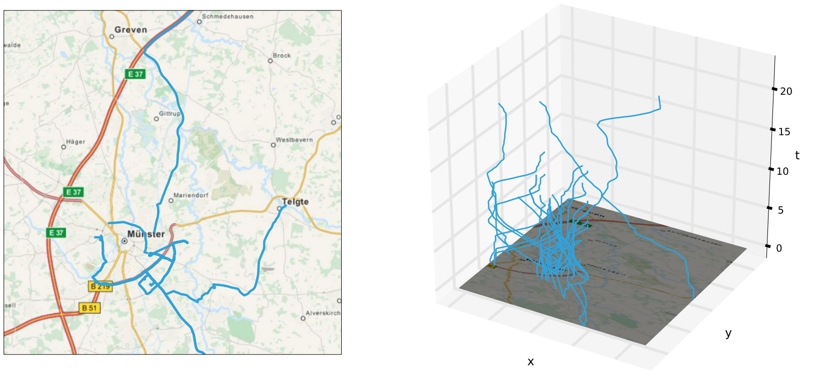
on computing joint potential path areas [44,45] for multiple space-time paths, is more complex, and requires multiple levels of queries and calculations. For these examples, we take advantage of vehicle trajectories from the EnviroCar[5] project's RESTful API. The (preliminary) Python code implementing the examples discussed in this section is available at `https://github.com/carsonfarmer/hypt`.

5.1 EnviroCar

EnviroCar is a community-based data collection platform for gathering vehicle-borne sensor data and producing environmental information [14]. EnviroCar uses standard Bluetooth OBD-II adapters[6], which are connected to a vehicle via the standard OBD connection that allows it to read parameters such as speed or revolutions per minute. From there, a smartphone records the data at regular time intervals, augmented with GPS information from the EnviroCar smartphone app. The EnviroCar app automatically calculates further information such as fuel consumption and CO_2 emissions, which can then be uploaded to the EnviroCar platform server for subsequent analysis and sharing with the wider research and citizen-science communities.

EnviroCar trajectories provide an ideal test-bed for exploring some of the concepts presented in this paper. For each control point in a series of EnviroCar

[5] `https://www.envirocar.org`
[6] `http://www.obdii.com/background.html`



(a) 2D trajectories (b) 3D trajectories

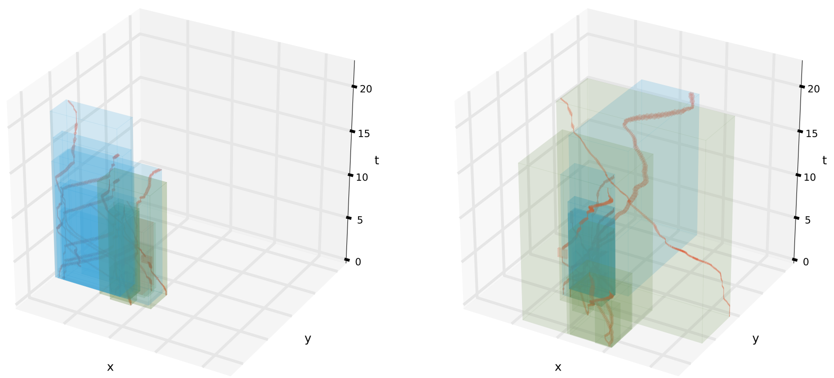
Fig. 4. A random selection of 43 EnviroCar space-time paths, encorporating ∼4050 space-time prisms (see Fig. 3b). Note that times have been scaled from 0 to ∼20 minutes for demonstration purposes. Basemap data, imagery, and map information provided by MapQuest, OpenStreetMap and contributors, ODbL. Trajectories data provided by EnviroCar [14], ODbL.

trajectories, we have several measures that can be used to determine the shape and size of its corresponding space-time prism. For instance, the recorded speed at each point in the trajectory can be used to determine the value of v_{max} (maximum velocity), which is of relevance when computing dynamic potential path areas or other metrics that are dependent on the space-time prism. Additional variables such as CO_2, can be stored along with the control point and associated prism to answer queries such as "how much CO_2 was produced by vehicles in this area over this time period?" or "which locations (joint potential path areas) have the highest number of CO_2 measurements in this region?".

5.2 Similarity Analysis

Similarity analysis across space-time paths is a common task in time geography research. The ability to identify similar space-time paths can aide researchers in locating space-time stations and bundling, improve visualization though path clustering (grouping similar paths), and path aggregation (forming composite paths) [41], as well as identifying similar geospatial 'lifelines' for discovering the environmental factors responsible for hot-spots and clusters of certain diseases [47]. Additionally, a common task in animal movement analysis is to identify areas of (potential) spatio-temporal overlap (or separation) between different animal species [48] or individual animals of the same species [45] (see Sec. 5.3). These types of analysis are generally aided by first identifying *similar* space-time paths.

A number of similarity measures exist in the literature (see [47], [41] for examples), and while it is not the goal of this paper to present a new comprehensive method for similarity analysis, frequently, the task of space-time path similarity search (or clustering) is a first step in an analytical workflow, designed to reduce complexity and aid pattern recognition. As such, the HPT framework presented here provides a useful heuristic for grouping similar space-time paths – with little to no additional effort on the part of the analyst. This is because the goal of the HPT algorithm is to minimize the size of the sub-volumes, and by doing so, they are also implicitly minimizing the 'distance' between space-time paths. Additionally, due to the incremental nature of the HPT update algorithm used here (see Fig. 2 for a discussion of tree restructuring via rotations), the addition of a new trajectory (or new control point in an existing trajectory) simply integrates with the existing trajectory 'clusters', and subsequent updates can potentially improve the optimality of the grouping over time.



(a) Left split of binary HPT (b) Right split of binary HPT

Fig. 5. Top-level split of a binary HPT (see Fig. 3) into left (a) and right (b) components. Within each split, the right and left components of the second-level split are denoted by different shading.

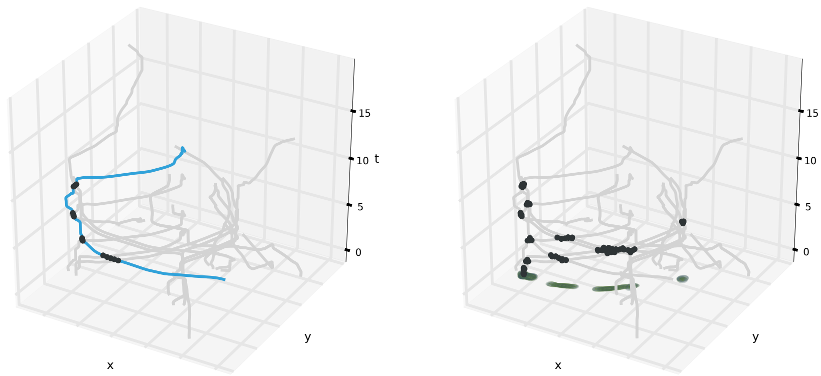
Figure 5 provides an example of the implicit 'grouping' of similar space-time paths using the previous EnviroCar trajectories example from Fig. 3: a top-level split of the tree into left (Fig. 5a) and right (Fig. 5b) components. This simple two-stage split separates trajectories into similar path-types, with paths circling Münster's downtown core in Fig. 5a and cross-/inter-town paths in Fig. 5b. Further similarity breakdowns can be observed, including two separate, temporally-offset, spatially-similar groupings in Fig. 5a with one showing travel between the University's geosciences building in the north-west and the Loddenheide area in the south (see Fig. 4 for reference).

5.3 Joint Potential Path Areas

In time geography analysis, it is often of interest to identity areas where interactions in space-time could occur. For example, researchers working with animal telemetry data may be interested in mapping regions where inter- or intra-species interactions may have occurred in an effort to better understand animal movement behaviors (avoidance, attraction, *etc.*) [44]. Similarly, it may be useful to highlight potential contact points for infectious disease transmission, or to identify regions of high or low densities of space-time interactions [49]. For the current example, we are interested in addressing the second question presented in Sec. 5.1, where we are trying to identify locations in the study region that have the highest number of CO_2 estimates. By determining these regions of overlap in space and time, we can potentially identify regions where we can have more confidence in our estimated CO_2 values.

To identify regions where multiple estimates have been made around the same space-time, we need to identify potential 'contact' points between vehicular trajectories, and then map their corresponding joint potential path areas (jPPA) [50]. A potential path area describes the elliptical region in space that a moving object or person could potentially reach given fixed start and end points. It can be conceptualized as the projection of the spacetime prism between two control points onto the geographical plane [3], [50]. As such, a jPPA is simply the 2D projection of the intersection of two space-time prisms. Previously, this type of analysis involved two steps: (1) determining potential space-time contacts by temporally syncing trajectories and performing distance-based queries at various time slices (space-time prisms can be used at the cost of additional computation), and (2) computing the intersection of identified prism-pairs at various time slices to compute the jPPA.

A naive version of the first step requires $\mathcal{O}(n^2)$ queries across a pair of trajectories, making it nearly impossible to scale to more than a handful of trajectories or control-points. Some efficiencies can be gained by using spatial indexing systems in a GIS-framework, however, this is often not done in practice. Because the HPT presented here is a binary tree (with a query time of $\mathcal{O}(\log n)$), we are able to reduce the time complexity of this process to $\mathcal{O}(n \log n)$ (additional speed gains are possible via more efficient 'dual-tree' approaches [51,52]). Figure 6a shows an example result for this type of query for a single trajectory to all other trajectories in a subset of the EnviroCar trajectories used previously. In this case, space-time contact is based on *potential* contact using the space-time prisms along the trajectories. Building on this, Fig. 6b shows potential contacts between *all* pairs of trajectories in the subset, along with their corresponding PPAs (projected onto the x/y plane). With the contact points identified, it is relatively straightforward to compute the relevant PPAs of the interacting space-time prisms by projecting their intersecting portions (portions that share the same space-time volume) onto the 2D geographic space. The jPPAs are then simply the geometric intersection of these PPAs (not shown), which can be computed using standard computational geometry techniques.



(a) Single-trajectory contacts (b) All-trajectories contacts

Fig. 6. Interaction patterns of a subset of the EnviroCar trajectories (see Fig. 5a), with potential contact points (PCP) for a single trajectory with all other trajectories in the subset (a), and the PCP between *all* trajectories in the subset and their corresponding PPAs. Note that we are showing overlapping PPAs (with darker regions representative of the jPPAs) that have been increased in size ($\times 5$) to aid in visualization.

6 Conclusions

The primary goal of this paper is to introduce methodological and technical improvements based on time-geographic theories and methods. To this end, we have presented an extensible framework for scaling time geographic methods to the increasingly large and diverse set of emerging spatio-temporal data sources. By taking advantage of techniques from the computer graphics literature, and combining these ideas in a time geography framework, we outline a hierarchical tree of space-time prisms, or Hierarchical Prism Tree (HPT), that forms the basis for a powerful computational framework for time geography research. In particular, our HPT is able to embed both space-time paths and prisms in a 3D space-time. This space-time tree is able to handle large volumes of space-time data that are potentially dynamic (and/or real-time) in nature. We demonstrated the utility of our approach using two common time geography analysis tasks, based on (1) space-time path similarity analysis, and (2) identifying joint potential path areas. While the work presented here is by no means exhaustive, it does provide a useful initial exploration of the utility of thinking about the scalability of time geographic methods. Indeed, the dynamic HPT presented in this paper provides an ideal framework for scaling and exploring time-geographic methods and ideas in an intuitive and computationally efficient manner.

The development of the HPT presented in this paper offers many avenues for further development. Currently, we are exploring ways to scale various space-time intersection queries in order to facilitate the data-driven generation of space-time prisms for data integration, as discussed in [42]. Additionally, the dynamic nature of the HPT is designed to facilitate tracking and analysis of real-time spatio-temporal data sources, such as those generated by the recently launched ICARUS initiative[7] or the long-established Argos system[8]. In order to make time geography methods accessible to the research communities working with such platforms, we are currently developing a suite of tools for working with space-time data using the Python programing language. Python is continuing to gain favor among data scientists and academic researchers, and implementation of various time geography methods within our HPT framework should facilitate increased adoption of time geography concepts and methods throughout the social and environmental sciences. A computational framework that is able to scale time geographic analysis from working with small, localized samples, to large, globally-distributed (possibly real-time) data sources has the potential to increase the utility of time geography concepts and methods to new domains and research questions significantly.

References

1. Batty, M.: Smart cities, big data. Environment and Planning B: Planning and Design **39**(2) (2012) 191–193
2. Yang, C., Raskin, R., Goodchild, M., Gahegan, M.: Geospatial cyberinfrastructure: Past, present and future. Computers, Environment and Urban Systems **34**(4) (2010) 264 – 277 Geospatial Cyberinfrastructure.
3. Miller, H.J.: A measurement theory for time geography. Geographical analysis **37**(1) (2005) 17–45
4. Hägerstrand, T.: What about people in regional science? Papers of the Regional Science Association **24** (1970) 7–21
5. Miller, H.J.: What about People in Geographic Information Science? In Fisher, P., Unwin, D., eds.: Re-presenting GIS. John Wiley & Sons, Ltd. (2005) 215–242
6. Shaw, S.L.: Guest editorial introduction: time geography – its past, present and future. Journal of Transport Geography **23**(0) (2012) 1 – 4 Special Issue on Time Geography.
7. Crease, P., Reichenbacher, T.: Linking Time Geography and Activity Theory to Support the Activities of Mobile Information Seekers. Transactions in GIS **17**(4) (2013) 507–525
8. Raubal, M., Miller, H.J., Bridwell, S.: User-centred time geography for location-based services. Geografiska Annaler: Series B, Human Geography **86**(4) (2004) 245–265
9. Kwan, M.P.: Gender and Individual Access to Urban Opportunities: A Study Using Space-Time Measures. The Professional Geographer **51**(2) (1999) 210–227
10. Miller, H.J.: Modelling accessibility using space-time prism concepts within geographical information systems. International Journal of Geographical Information System **5**(3) (1991) 287–301

[7] ICARUS analyzes the migratory behavior of animals such as birds and bats: `http://icarusinitiative.org`

[8] Argos is a global, satellite-based platform widely used in animal tracking: `http://www.argos-system.org/`

11. Raubal, M., Winter, S., Teßmann, S., Gaisbauer, C.: Time geography for ad-hoc shared-ride trip planning in mobile geosensor networks. ISPRS Journal of Photogrammetry and Remote Sensing **62**(5) (2007) 366–381
12. Winter, S., Raubal, M.: Time Geography for Ad-Hoc Shared-Ride Trip Planning. In: 7th International Conference on Mobile Data Management, 2006. MDM 2006. (2006)
13. Rainham, D., McDowell, I., Krewski, D., Sawada, M.: Conceptualizing the healthscape: Contributions of time geography, location technologies and spatial ecology to place and health research. Social Science & Medicine **70**(5) (2010) 668 – 676
14. Bröring, A., Remke, A., Stasch, C., Autermann, C., Rieke, M., Möllers, J.: EnviroCar: A Citizen Science Platform for Analyzing and Mapping Crowd-Sourced Car Sensor Data. Transactions in GIS **19**(3) (2015) 362–376
15. Winter, S., Yin, Z.C.: The elements of probabilistic time geography. GeoInformatica **15**(3) (2011) 417–434
16. Samet, H.: Applications of Spatial Data Structures. Addison-Wesley (1990)
17. Myllymaki, J., Kaufman, J.: High-performance spatial indexing for location-based services. In: Proceedings of the 12th International Conference on World Wide Web. WWW '03, New York, NY, USA, ACM (2003) 112–117
18. Gustafsson, T., Hansson, J.: Dynamic on-demand updating of data in real-time database systems. In: Proceedings of the 2004 ACM Symposium on Applied Computing. SAC '04, New York, NY, USA, ACM (2004) 846–853
19. Papadias, D., Tao, Y., Kanis, P., Zhang, J.: Indexing spatio-temporal data warehouses. In: Data Engineering, 2002. Proceedings. 18th International Conference on. (2002) 166–175
20. Theodoridis, Y., Sellis, T., Papadopoulos, A., Manolopoulos, Y.: Specifications for efficient indexing in spatiotemporal databases. In: Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on. (Jul 1998) 123–132
21. Wang, W., Yang, J., Muntz, R.: Pk-tree: A spatial index structure for high dimensional point data. In Tanaka, K., Ghandeharizadeh, S., Kambayashi, Y., eds.: Information Organization and Databases. Volume 579 of The Springer International Series in Engineering and Computer Science. Springer US (2000) 281–293
22. Tayeb, J., Ulusoy, Ö., Wolfson, O.: A quadtree-based dynamic attribute indexing method. The Computer Journal **41**(3) (1998) 185–200
23. Navarro, G., Reyes, N.: Dynamic spatial approximation trees for massive data. In: Second International Workshop on Similarity Search and Applications. SISAP (Aug 2009) 81–88
24. Navarro, G., Reyes, N.: Dynamic spatial approximation trees. J. Exp. Algorithmics **12** (June 2008) 1.5:1–1.5:68
25. Bo, Z., Fu-ling, B.: Dynamic quadtree spatial index algorithm for mobile gis. Computer Engineering **33**(15) (2007) 86
26. Xia, Y., Prabhakar, S.: Q+rtree: efficient indexing for moving object databases. In: Database Systems for Advanced Applications, 2003. (DASFAA 2003). Proceedings. Eighth International Conference on. (March 2003) 175–182
27. Myllymaki, J., Kaufman, J.: Dynamark: A benchmark for dynamic spatial indexing. In Chen, M.S., Chrysanthis, P.K., Sloman, M., Zaslavsky, A., eds.: Mobile Data Management. Volume 2574 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2003) 92–105
28. Myllymaki, J., Kaufman, J.: Locus: A testbed for dynamic spatial indexing. IEEE Data Engineering Bulliten Special Issue On Indexing Of Moving Objects **25** (2002) 48–55

29. Zhu, Q., Gong, J., Zhang, Y.: An efficient 3d r-tree spatial index method for virtual geographic environments. Journal of Photogrammetry and Remote Sensing **62**(3) (2007) 217 – 224
30. Ize, T., Wald, I., Parker, S.G.: Asynchronous bvh construction for ray tracing dynamic scenes on parallel multi-core architectures. In: Proceedings of the 7th Eurographics Conference on Parallel Graphics and Visualization. EGPGV '07, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2007) 101–108
31. Glassner, A.S.: An introduction to ray tracing. Academic Press Ltd., London, UK, UK (1989) 1–31
32. Stich, M., Friedrich, H., Dietrich, A.: Spatial splits in bounding volume hierarchies. In: Proceedings of the Conference on High Performance Graphics 2009. HPG '09, New York, NY, USA, ACM (2009) 7–13
33. Maneewongvatana, S., Mount, D.M.: Analysis of approximate nearest neighbor searching with clustered point sets. CoRR **cs.CG/9901013** (1999)
34. Vinkler, M., Havran, V., Bittner, J.: Bounding volume hierarchies versus kd-trees on contemporary many-core architectures. In: Proceedings of the 30th Spring Conference on Computer Graphics. SCCG '14, New York, NY, USA, ACM (2014) 29–36
35. Shevtsov, M., Soupikov, A., Kapustin, A.: Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes. Computer Graphics Forum **26**(3) (2007) 395–404
36. He, L., Ortiz, R., Enquobahrie, A., Manocha, D.: Interactive continuous collision detection for topology changing models using dynamic clustering. In: Proceedings of the 19th Symposium on Interactive 3D Graphics and Games. i3D '15, New York, NY, USA, ACM (2015) 47–54
37. Stein, C., Limper, M., Kuijper, A.: Spatial data structures for accelerated 3d visibility computation to enable large model visualization on the web. In: Proceedings of the 19th International ACM Conference on 3D Web Technologies. Web3D '14, New York, NY, USA, ACM (2014) 53–61
38. Kopta, D., Ize, T., Spjut, J., Brunvand, E., Davis, A., Kensler, A.: Fast, effective BVH updates for animated scenes. In: Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. I3D '12, New York, NY, USA, ACM (2012) 197–204
39. Yoon, S.E., Curtis, S., Manocha, D.: Ray tracing dynamic scenes using selective restructuring. In: Proceedings of the 18th Eurographics Conference on Rendering Techniques. EGSR'07, Aire-la-Ville, Switzerland, Switzerland, Eurographics Association (2007) 73–84
40. Karras, T., Aila, T.: Fast parallel construction of high-quality bounding volume hierarchies. In: Proceedings of the 5th High-Performance Graphics Conference. HPG '13, New York, NY, USA, ACM (2013) 89–99
41. Miller, H., Raubal, M., Jaegal, Y.: Measuring space-time prism similarity through temporal profile curves. In: 19th AGILE Conference on Geographic Information Science – Geospatial Data in a Changing World. (2016) 19
42. Keßler, C., Farmer, C.J.Q.: Querying and integrating spatial–temporal information on the web of data via time geography. Web Semantics: Science, Services and Agents on the World Wide Web **35**(1) (2015) 25–34
43. Schwesinger, U., Siegwart, R., Furgale, P.: Fast collision detection through bounding volume hierarchies in workspace-time space for sampling-based motion planners. In: Robotics and Automation (ICRA), 2015 IEEE International Conference on. (May 2015) 63–68

44. Long, J., Nelson, T.: Home range and habitat analysis using dynamic time geography. The Journal of Wildlife Management **79**(3) (2015) 481–490
45. Long, J.A., Nelson, T.A.: Measuring dynamic interaction in movement data. Transactions in GIS **17**(1) (2013) 62–77
46. Larsson, T., Akenine-Möller, T.: A dynamic bounding volume hierarchy for generalized collision detection. Computers & Graphics **30**(3) (2006) 450 – 459
47. Sinha, G., Mark, D.M.: Measuring similarity between geospatial lifelines in studies of environmental health. Journal of Geographical Systems **7**(1) (2005) 115–136
48. Gao, P., Kupfer, J.A., Zhu, X., Guo, D.: Quantifying animal trajectories using spatial aggregation and sequence analysis: A case study of differentiating trajectories of multiple species. Geographical Analysis (2016) n/a–n/a
49. Demšar, U., Virrantaus, K.: Space–time density of trajectories: exploring spatio-temporal patterns in movement data. International Journal of Geographical Information Science **24**(10) (2010) 1527–1542
50. Long, J.A., Webb, S.L., Nelson, T.A., Gee, K.L.: Mapping areas of spatial-temporal overlap from wildlife tracking data. Movement Ecology **3**(1) (2015) 1–14
51. Ram, P., Lee, D., March, W., Gray, A.G.: Linear-time Algorithms for Pairwise Statistical Problems. In: Advances in Neural Information Processing Systems (NIPS) 22 (Dec 2009), MIT Press (2010)
52. Gray, A.G., Moore, A.W.: *N*-Body Problems in Statistical Learning. In Leen, T.K., Dietterich, T.G., Tresp, V., eds.: Advances in Neural Information Processing Systems (NIPS) 13 (Dec 2000), MIT Press (2001)