2019 IEEE 35th International Conference on Data Engineering (ICDE)

Continuous Range Queries over Multi-Attribute Trajectories

Jianqiu Xu[1] Zhifeng Bao[2] Hua Lu[3]
Nanjing University of Aeronautics and Astronautics[1], China
RMIT University[2], Australia
Aalborg University[3], Denmark
jianqiu@nuaa.edu.cn, zhifeng.bao@rmit.edu.au, luhua@cs.aau.dk

***Abstract*—A multi-attribute trajectory consists of a sequence of time-stamped locations and a set of attributes that characterize diverse aspects of the corresponding moving object. In this paper, we study continuous range queries over multi-attribute trajectories. Such a query returns the objects whose attributes contain expected values and whose locations are always within a distance threshold to the query trajectory during the entire overlapping time period. To efficiently answer the query, an optimal method of partitioning the trajectories is proposed and an index structure is developed to support the combined search of spatio-temporal parameters and attribute values. We provide a general solution that is able to process multi-attribute trajectories as well as traditional trajectories without attributes. We carry out comprehensive experiments in a prototype database system to evaluate the efficiency and scalability of our designs. The experimental results show that our approach outperforms five alternative approaches by a factor of 5-50x on large datasets.**

I. Introduction

The increasing prevalence of GPS-equipped mobile devices has led to an explosion of *spatio-temporal trajectories*. In the last decade, a rich body of research has been conducted on processing spatio-temporal trajectories [3], [17], [15]. In practice, objects/entities are naturally of multiple attributes in addition to spatio-temporal aspects [10], [18], amenable to diverse types of analysis. An important problem in the context of applications that leverage multiple attributes is how to efficiently find objects by queries formulated as a combination of spatio-temporal and attribute constraints. The goal of this paper is to develop novel query processing capabilities to address spatio-temporal trajectories associated with descriptive attributes for their corresponding moving objects, called *multi-attribute trajectories*.

Consider a motivation scenario in Figure 1. There are four vehicle trajectories, each of which contains two attributes COLOR and BRAND with domains SILVER, RED and BMW, VW, VOLVO, respectively. Object o_3 is a special object that carries VIP passengers or sensitive materials. For security reasons, it is needed to detect whether the special object is stalked. To this end, we make use of the multiple attributes to form a semantic-richer query, e.g., *Is any SILVER VW always kept within 50 meters to o_3?* Such a query is called continuous range query with attributes, CRA for short. The returned objects must satisfy the criteria: (i) *time-dependent distance constraint* and (ii) *attribute consistency*. The CRA query is essential for traffic monitoring applications, but to the best of our knowledge it has not been studied before.

It is noteworthy that spatio-temporal trajectory databases are only suitable for finding the objects fulfilling spatio-temporal conditions, and thus fall short for the practical needs described above. In the example, although o_1 is within 50 meters to o_3 for a while, it is not a SILVER VW and should not be returned. On the other hand, traditional range queries evaluate objects at a given time point. In contrast, we consider the continuous version that the objects should be always within the distance to the target during the overlapping time. This complicates the evaluation because objects within the query distance may change over time. Objects o_4 and o_2 fulfill the condition during $[t_1, t_2]$ and $[t_2, t_3]$, respectively, but there is no (SILVER, VW) following o_3 during o_3's timespan. As a result, no stalker is detected.

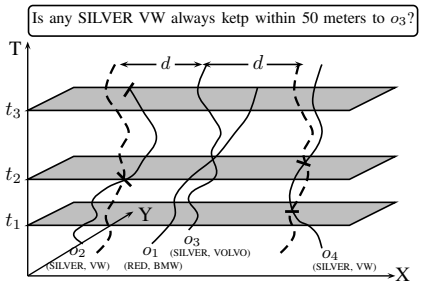


Fig. 1. Example of CDA

In the literature, several studies have investigated spatio-temporal trajectories with supplementary data, called *semantic trajectories* [4], [2], [14]. We compare them as follows: (i) Semantic trajectories enrich spatio-temporal trajectories by virtue of keywords or semantic labels that are related to locations such as points of interest, whereas multi-attribute trajectories consider location-independent characteristics. (ii) Semantic locations are sparsely defined as only a few locations of the trajectory may have semantics. For example, given a semantic trajectory $o'_1 = \langle(loc_1, t_1, \textit{coffee}), (loc_2, t_2, \textit{pizza})\rangle$, the meaning is clear at locations loc_1 and loc_2 but no semantic is defined between loc_1 and loc_2 . Our attributes are associated with the complete trajectory. (iii) Distinct queries are processed. Queries in semantic trajectories incorporate the measurement of spatial and textual relevances in order to find the most relevant trajectories, e.g., *ranked retrieval* and *top-k retrieval*. In contrast, we deal with the continuous evaluation of spatio-temporal ranges and the exact match on attributes.

Efficient data management requires underlying systems to

2375-026X/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDE.2019.00154

be complemented in terms of data representation and indexing methods. To these ends, we model attributes and integrate them with spatio-temporal trajectories into a unified framework. We adapt a standard 3-D R-tree by deploying an optimal partitioning of spatio-temporal trajectories. The goal is to normalize trajectories to create an R-tree with a good shape. An attribute structure is created on top of the R-tree to maintain attribute values. We design a flexible method such that the attribute structure can be discarded if only spatio-temporal trajectories are processed.

The contributions of the paper are summarized as follows: (i) We represent multi-attribute trajectories and formulate the continuous range query over such trajectories; (ii) An index structure built on an optimal partition of trajectories is proposed to support the search of both spatio- temporal parameters and attributes; (iii) We develop efficient query algorithms with effective pruning techniques and search heuristics. (iv) A thorough experimental study is performed using real GPS records and synthetic attribute values. The experimental results demonstrate that our approach outperforms five alter- natives by a factor of 5-50x on large datasets.

The rest of the paper is organized as follows. In Section II, we review the related work. The problem is defined in Section III. The index structure and query algorithms are proposed in Sections IV and V, respectively. We perform the evaluation in Section VI, followed by conclusions in Section VII.

II. Related work

Queries of spatio-temporal trajectories find objects from the spatio-temporal aspect, e.g., similar trajectories [17], nearest neighbors [13] and prediction [12]. Emerging applications require extensive information about movement data such as quality and semantics [22]. A semantic enriched trajectory is typically defined as a sequence of time-stamped places, each of which is represented by a location with a semantic label. Extracting semantic behavior from spatio-temporal trajectories is investigated [20] by identifying object stops or moves and annotate relevant locations with semantics such as *market* or *office*.

Attaching semantic labels to trajectories enables queries and analytics considering semantic interests and location preferences. Existing work falls into three categories: (i) *Ranked* and *top-k retrieval*. Relevant queries consider actions/activities that users can take at particular places such as *sport* and *dining*. A *top-k exemplar trajectory* query [14] consists of a set of locations with keywords and aims to find the most relevant trajectories in terms of the spatial and textual similarity. (ii) *Data mining and analytics*. Frequent sequential patterns can be found to reflect movement regularity by considering spatial compactness, semantic consistency and temporal continuity simultaneously [21]. A regional semantic trajectory pattern mining problem is also studied [2]. (iii) Range queries on multi-attribute trajectories return trajectories that contain query attributes and intersect a spatio-temporal window [19].

A systematic study is performed to capture a wide range of meanings related to locations including street names, transportation modes and speed profile [7]. A time-dependent label is defined to represent so-called *symbolic trajectories*, but time-dependent locations are not included in the model.

III. Problem definition

Let A be the set of multiple attributes. The ith attribute and its domain are denoted by $A[i]$ and $dom(A[i])$ ($i \in 1,..., |A|$), respectively. We assume that each $dom(A[i])$ is represented by a set of positive integers and define a data type $D_{\underline{att}}$ for the set of multiple attributes. For the sake of readability, the enum type is used for attributes COLOR and BRAND in Figure 1.

Definition 1 *The multi-attribute representation*

$D_{\underline{att}} = \{(a_1, ..., a_{|A|}) | \ a_i \in dom(A[i]), \ i \in \{1, ..., |A|\}\}$ *such that (i)* $\forall i \in \{1,..., |A|\}$*:* $dom(A[i]) \subset N^+$*; (ii)* $\forall \ i, j \in \{1,..., |A|\}$*:* $i \neq j \Rightarrow dom(A[i]) \cap dom(A[j]) = \varnothing$.

Let $\mathcal{O}$ be a set of multi-attribute trajectories. Each $o \in \mathcal{O}$ is denoted by o(*Trip*, *Att*) in which o.*Trip* and o.*Att* refer to a spatio-temporal trajectory and attributes, respectively. A spatio-temporal trajectory is represented by a data type *mpoint* [8]. Table I gives the representation of example trajectories.

TABLE I
AN EXAMPLE OF MULTI-ATTRIBUTE TRAJECTORIES

Id: *int*	*Trip*: *mpoint*	*Att*: *att*
o_1	location+time	(RED, BMW)
o_2	location+time	(SILVER, VW)
o_3	location+time	(SILVER, VOLVO)
o_4	location+time	(SILVER, VW)

Let $T(o)$ return the time period of a trajectory. We employ the function in [5] to return the time-dependent distance between two trajectories $o_1, o_2 \in \mathcal{O}$, denoted by $dist(o_1, o_2, T(o_1) \cap T(o_2))$. The query predicate Q_a defines a component for each attribute. Let $Q_a[j] \in dom(A[j]) \cup \{\perp\}$ refer to the jth attribute value. We define an operator called **contain**(o.*Att*, Q_a) that returns *true* if $\forall \ Q_a[j] \neq \perp$: o.*Att*[j] = $Q_a[j]$. The studied query CDA is formulated below.

Definition 2 ***Continuous distance queries with attributes***

Given a query trajectory o_q*, threshold* d *and attribute predicate* Q_a*, CDA returns* $\mathcal{O}' \subseteq \mathcal{O}$ *such that* $\forall \ o' \in \mathcal{O}'$*:* (i) ***contain***$(o'.Att, Q_a)$; $(ii) \forall \ t \in T(o_q) \cap T(o')$, $dist(o_q, o', t) < d$.

Referring to Figure 1, the CRA query finds o_4 at $[t_1, t_2]$ and o_2 at $[t_2, t_3]$, but no object fulfills the condition during $[t_1, t_3]$. Table II lists the notations frequently used in the paper.

TABLE II
NOTATIONS

Notation	Description
$\mathcal{O}$	the multi-attribute trajectory database
o	a multi-attribute trajectory
$\|A\|$	the number of attributes
$dom(A[i])$, $dom(A)$	the domain of $A[i]$, the overall domain
o_q, d	a query trajectory, the query distance
Q_a	the query attribute
t, $T(o)$	a time point, the time period of a trajectory

IV. The index structure

We design an index structure named *GR^2-tree* including two components: GR-tree and R_{att}. The GR-tree is an adapted 3-D R-tree built on partitioned spatio-temporal trajectories, and R_{att} is a relation for managing attribute values.

A. GR-tree

Partitioning trajectories. We normalize spatio-temporal trajectories to create the R-tree with a good shape. The time dimension is partitioned into a set of equal-sized intervals $\{T_1,...,T_K\}$ ($K > 1$) and the 2-D space is partitioned into a set of equal-sized cells. Given a multi-attribute trajectory, its spatio-temporal trajectory is split into a set of so-called *cell trajectories*, each of which represents the movement within a cell during an interval $T_k \in \{T_1 ,...,T_K\}$.

Definition 3 *Cell trajectory*

Let Cell(o, t) return the cell where o is located at a time point $t \in T(o)$. A cell trajectory $o[i]$ is a subset of o.Trip such that (i) $\forall\ t_1, t_2 \in T(o[i])$: Cell($o[i]$, t_1) = Cell($o[i]$, t_2); (ii) $\exists\ T_k \in \{T_1,..., T_K\}$: $T(o[i]) \subseteq T_k$.

For each $o \in \mathcal{O}$, we partition *o.Trip* into pieces according to time and identify the cells intersecting each piece. We may encounter the case that several cell trajectories are located in one cell. This means that the object enters the cell more than once. The GR-tree is built by bulk loading on cell trajectories sorted by time, cell id and 3-D bounding box. In order to preserve the spatio-temporal proximity, each leaf node only maintains cell trajectories with the same time interval and the same cell id. Each GR-tree node is supplemented by a bitmap representing the cells intersecting with the 2-D bounding box of the node.

B. The attribute structure

The relation R_{att} records attribute values of multi-attribute trajectories maintained in GR-tree nodes. The attribute values for a leaf node are obtained by accessing the underlying data and the values for a non-leaf node are obtained by performing the union on values of its child nodes. Each tuple in R_{att} is of the form (*nid*, *a_tr*, *b*) with respect to an attribute value contained by the node, in which *nid* is a node id, *a_tr* is a transformed attribute value and *b* is a bitmap. The transformed value is uniquely achieved by interleaving the binary representation of the attribute id and the value. We create a B-tree on R_{att} by combining *nid* and *a_tr* as the key. The bitmap records the entries containing the attribute value, enabling us to only access qualified entries instead of performing a sequential scan. We perform the mapping between the size of the bit array $|b|$ and the maximum number of entries in a GR-tree node f (i.e., the *fan-out*): (i) $|b| \geq f$, each bit maps to a unique entry. If the ith $\in [0,\ f)$ entry contains the value, we have $b[i] = 1$. Otherwise, $b[i] = 0$. (ii) $|b| < f$, each bit maps to a range of entries and entries for the ith bit are calculated by $[i \cdot \lceil \frac{f}{|b|} \rceil, (i+1) \cdot \lceil \frac{f}{|b|} \rceil]$. We define $b[i] = 1$ if one of the entries contains the value.

Example. We report the attribute relation by referring to Figure 2. Let N denote a GR-tree node. The bitmap 00001001 in the first row represents the 1st and 4th entries in N_r containing RED, i.e., N_1 and N_4.

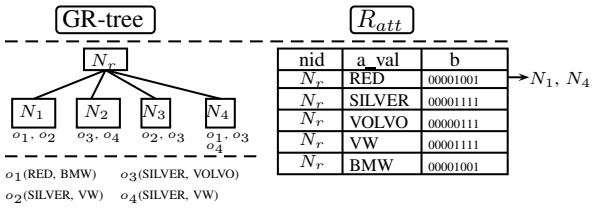


Fig. 2. Exemplify the GR^2-tree

V. Query procedure

Employing the GR^2-tree, we answer the query in three steps, as illustrated in Figure 3. Step 1 establishes the spatio-temporal area restricted by o_q and d, which is represented by a set of time-dependent cells. Step 2 performs a breadth-first traversal on the GR^2-tree to prune the search space by taking into account both spatio-temporal parameters and attribute values. Given a GR-tree node, we retrieve its cell bitmap and determine the cells intersecting the node. The node can be pruned if there is no overlap between the cells intersecting the node and the query. We return a set of candidates, each of which is a cell trajectory that (i) contains Q_a and (ii) has the distance to o_q less than d. The distance is an approximate value calculated by using minimum bounding boxes of trajectories. A candidate is marked if its maximum distance to o_q is less than d. Step 3 iteratively checks the accurate distance between each candidate and the query. If the candidate is *marked*, we directly put it into the result set. Otherwise, the actual distance is computed. A trajectory may be split because only the piece of movements fulfilling the distance condition is considered.

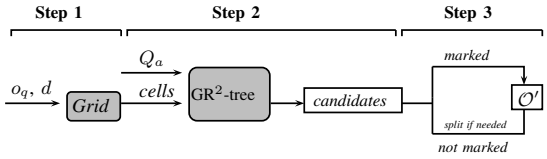


Fig. 3. Three steps of answering the query

VI. Experimental evaluation

We implement the proposed methods in C/C++ and perform the evaluation in an extensible database system SECONDO [6]. A standard PC (Intel(R) Core(TM) i7-4770CPU, 3.4GHz, 4GB memory, 2TB hard disk) running Suse Linux 13.1 (32 bits, kernel version 3.11.6) is used.

Datasets and parameters. We use real GPS records of Beijing taxis [1], named BTAXI. We develop a tool to generate attributes. Table III reports the dataset statistics and parameter settings. The CPU time and I/O accesses are used as performance metrics and the results are averaged over 20 runs.

Baseline methods: 1) **3-D R-tree**; (2) **RIB**, we adapt the method in [16] that that groups multi-attribute trajectories by

TABLE III
DATASETS AND PARAMETER SETTINGS

Name	#GPS Records	$\|\mathcal{O}\|$	$\|A\|$	$dom(A)$	X and Y ranges
BTAXI	235,634,511	4,220,435	10	[1, 151]	[21, 119958], [0, 119653]
Query settings					
$\|Q_a\|$: 3			d (km): 10		

attribute values and employs an inverted bitmap; (3) **4-D R-tree**; (4) **IOC-Tree** [9]; (5) **HAGI** [11].

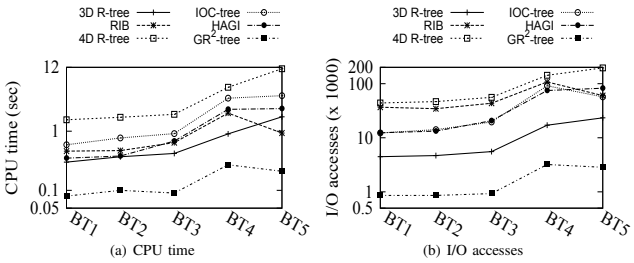
Scaling the number of trajectories. To vary the data size, different subsets of BTAXI are selected, as summarized in Table IV. The performance result is reported in Figure 4. When the data size grows the costs of all methods rise proportionally, but our method outperforms baseline methods by a factor of 5-50x on the largest dataset.

TABLE IV
DATASETS FOR SCALING $|\mathcal{O}|$, $|A|$ AND DOM(A)

Name	$\|\mathcal{O}\|$	$\|A\|$	$dom(A)$
BT1	533,635	10	[1, 151]
BT2	1,009,579		
BT3	1,424,273		
BT4	2,757,312		
BT5	4,220,435		

$\|A\|$	$dom(A)$
1	[1, 5]
2	[1, 43]
5	[1, 74]
10	[1, 211]
15	[1, 322]

$\|A\|$	$dom(A)$
1	[1, 5]
1	[1, 20]
1	[1, 50]
1	[1, 100]
1	[1, 200]



(a) CPU time (b) I/O accesses

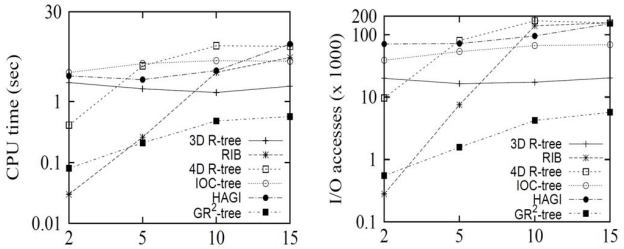
Fig. 4. Scaling $|\mathcal{O}|$

Scaling data attributes. To investigate the effect of attributes on the performance, we choose the largest number of trajectories and vary the attribute setting by scaling (i) the number of attributes and (ii) the domain, as reported in Table IV. The performance results are reported in Figure 5. Our method is superior than other methods when the number of attributes increases. RIB is slightly better than our solution when $|A|$ = 1 and $|A|$ = 2. This is because the index is built on objects grouped by attribute values and a good locality is achieved in terms of attributes. The behavior also occurs when $|A|$ = 1. However, RIB's performance degrades significantly when $|A|$ becomes large. It is a non-trivial task to group multi-attribute objects to build the index for RIB.

VII. CONCLUSIONS

We studied multi-attribute trajectories and proposed a new query. An index structure as well as efficient query algorithms were developed. Extensive experimental results demonstrated that our method significantly outperforms alternative methods.

Acknowledgment. This work is supported by National Key Research and Development Plan of China (2018YFB1003902), the Fundamental Research Funds for the Central Universities, NO. NS2017073, ARC DP170102726, DP180102050, and NSFC 61728204, 91646204.



(a) CPU time (b) I/O accesses

Fig. 5. Scaling $|A|$

REFERENCES

[1] http://factory.datatang.com/en/ (2017).
[2] D.W. Choi, J. Pei, and T. Heinis. Efficient mining of regional movement patterns in semantic trajectories. *PVLDB*, 10(13):2073–2084, 2017.
[3] J. Dai, B. Yang, C. Guo, C. S. Jensen, and J. Hu. Path cost distribution estimation using trajectory data. *PVLDB*, 10(3):85–96, 2016.
[4] M. Luisa Damiani and R. H. Güting. Semantic trajectories and beyond. In *IEEE MDM*, pages 1–3, 2014.
[5] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis. Algorithms for nearest neighbor search on moving object trajectories. *GeoInformatica*, 11(2):159–193, 2007.
[6] R. H. Güting, T. Behr, and C. Düntgen. SECONDO: A platform for moving objects database research and for publishing and integrating research implementations. *IEEE Data Eng. Bull.*, 33(2):56–63, 2010.
[7] R. H. Güting, F. Valdés, and M.L. Damiani. Symbolic Trajectories. *ACM Transactions on Spatial Algorithms and Systems*, 1(2):Article 7, 2015.
[8] R.H. Güting, M.H. Böhlen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M. Schneider, and M. Vazirgiannis. A foundation for representing and querying moving objects. *ACM TODS*, 25(1):1–42, 2000.
[9] Y. Han, L. Wang, Y. Zhang, W. Zhang, and X. Lin. Spatial keyword range search on trajectories. In *DASFAA*, pages 223–240, 2015.
[10] G. Li, J. He, D. Deng, and J. Li. Efficient similarity join and search on multi-attribute data. In *ACM SIGMOD*, pages 1137–1151, 2015.
[11] Y. Su, Y. Wu, and A. L. P. Chen. Monitoring heterogeneous nearest neighbors for moving objects considering location-independent attributes. In *DASFAA*, pages 300–312, 2007.
[12] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, and W. Lv. The simpler the better: A unified approach to predicting original taxi demands based on large-scale online platforms. In *ACM SIGKDD*, pages 1653–1662, 2017.
[13] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, and G. Cong. Reverse k nearest neighbor search over trajectories. *IEEE Trans. Knowl. Data Eng.*, 30(4):757–771, 2018.
[14] S. Wang, Z. Bao, J. S. Culpepper, T. Sellis, M. Sanderson, and X. Qin. Answering top-k exemplar trajectory queries. In *ICDE*, pages 597–608, 2017.
[15] S. Wang, Z. Bao, J. Shane Culpepper, Z. Xie, Q. Liu, and X. Qin. Torch: A search engine for trajectory data. In *ACM SIGIR*, pages 535–544, 2018.
[16] D. Wu, M. L. Yiu, G. Cong, and C. S. Jensen. Joint top-k spatial keyword query processing. *IEEE Trans. Knowl. Data Eng.*, 24(10):1889–1903, 2012.
[17] D. Xie, F. Li, and J. M. Phillips. Distributed trajectory similarity search. *PVLDB*, 10(11):1478–1489, 2017.
[18] J. Xu, D. V. Kalashnikov, and S. Mehrotra. Efficient summarization framework for multi-attribute uncertain data. In *SIGMOD*, pages 421–432, 2014.
[19] J. Xu, H. Lu, and R. H. Güting. Range queries on multi-attribute trajectories. *IEEE Trans. Knowl. Data Eng.*, 30(6):1206–1211, 2018.
[20] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Semitri: a framework for semantic annotation of heterogeneous trajectories. In *EDBT*, pages 259–270, 2011.
[21] C. Zhang, J. Han, L. Shou, J. Lu, and T. F. La Porta. Splitter: Mining Fine-Grained Sequential Patterns in Semantic Trajectories. *PVLDB*, 7(9):769–780, 2014.
[22] K. Zheng and H. Su. Go beyond raw trajectory data: Quality and semantics. *IEEE Data Eng. Bull.*, 38(2):27–34, 2015.