


Article

An Efficient Graph-Based Spatio-Temporal Indexing Method for Task-Oriented Multi-Modal Scene Data Organization

Bin Feng ¹, Qing Zhu ^{1,*}, Mingwei Liu ¹, Yun Li ¹ , Junxiao Zhang ¹, Xiao Fu ¹, Yan Zhou ^{2,*}, Maosu Li ¹, Huagui He ³ and Weijun Yang ³

¹ Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 610031, China; bk20090770@my.swjtu.edu.cn (B.F.); liumingwei@my.swjtu.edu.cn (M.L.); liyun20151202@outlook.com (Y.L.); zjx-gis-swjtu@foxmail.com (J.Z.); cathy_fuxiao@163.com (X.F.); limaosu_gis@my.swjtu.edu.cn (M.L.)

² School of Resource and Environment, University of Electric Science and Technology, Chengdu 611731, China

³ Guangzhou Urban Planning & Design Survey Research Institute, Guangzhou 510060, China; gzhehuagui@163.com (H.H.); fazeyang@gmail.com (W.Y.)

* Correspondence: zhuq66@263.net (Q.Z.); zhouyan_gis@uestc.edu.cn (Y.Z.)

Received: 13 July 2018; Accepted: 4 September 2018; Published: 8 September 2018



Abstract: Task-oriented scene data in big data and cloud environments of a smart city that must be time-critically processed are dynamic and associated with increasing complexities and heterogeneities. Existing hybrid tree-based external indexing methods are input/output (I/O)-intensive, query schema-fixed, and difficult when representing the complex relationships of real-time multi-modal scene data; specifically, queries are limited to a certain spatio-temporal range or a small number of selected attributes. This paper proposes a new spatio-temporal indexing method for task-oriented multi-modal scene data organization. First, a hybrid spatio-temporal index architecture is proposed based on the analysis of the characteristics of scene data and the driving forces behind the scene tasks. Second, a graph-based spatio-temporal relation indexing approach, named the spatio-temporal relation graph (STR-graph), is constructed for this architecture. The global graph-based index, internal and external operation mechanisms, and optimization strategy of the STR-graph index are introduced in detail. Finally, index efficiency comparison experiments are conducted, and the results show that the STR-graph performs excellently in index generation and can efficiently address the diverse requirements of different visualization tasks for data scheduling; specifically, the STR-graph is more efficient when addressing complex and uncertain spatio-temporal relation queries.

Keywords: multi-modal scene data; graph-based index; spatio-temporal relation query; visualization task; data organization

1. Introduction

High-performance scene data organization is one of the most common and fundamental functions of visualization in the field of geographic information systems (GIS) in the era of big data and cloud computing for smart cities [1,2]. With the development of social networks, sensor networks, and the Internet of Things (IOT), scene data organization in GIS has expanded into three spaces: cyber, physical, and social [3–5]. Hybrid spatio-temporal index methods based on trees or grids have been widely studied in data scheduling and visualization. Most of the existing spatio-temporal indexing methods index time and spatial dimension information and aim at maximizing input/output (I/O) of the data [6,7]. They cannot efficiently organize and manage the associated dynamic multi-modal scene data for diverse scene tasks, or respond to complex spatio-temporal relation queries in real

time. The organization of dynamic data (e.g., the flow of people, cars, air, and goods) and associated data (e.g., social, transportation, and semantic networks) has become a popular issue in the scene data organization domain [8]. Scene data organization is different from the data organization on the database side. It is task-oriented and requires high-performance responses to diverse query requirements in real-time, such as computation, visualization, and analysis [9,10]. Scene data organization and management methods are affected by multi-modal scene features and multi-level scene visualization task connotations. On the one hand, the time-varying and increasing nature of dynamic data and the complex, heterogeneous features of associated data have put forward new challenges for the organization and management of scene data, especially for spatio-temporal data indexing algorithms [11,12]. On the other hand, the I/O-intensive, computing-intensive, and interaction-intensive features of multi-level visualization tasks have focused on the development of new spatio-temporal indexes and query algorithms to support the real-time generation and interaction of scenes [13]. Therefore, how to deal with the I/O-intensive, computing-intensive, and interaction-intensive problems caused by multi-level scene tasks efficiently are the great challenges for scene data indexing methods.

Existing indexing methods for scene data organization can be classified into three groups: indexes for I/O-intensive applications, indexes for computing-intensive applications, and indexes for interaction-intensive applications. The indexes in the first group are designed to load data quickly and precisely, and there are two types of indexes. The first involves tree-based indexing methods, such as R-tree, Quad-tree, and their variants, which are based on external storage and are commonly used in traditional spatial database systems [14]. Traditional spatial database systems have been empowered with rich functionalities by using tree-like structure indexes to support efficient multi-dimensional access, but they cannot scale well enough to address large volumes of data and support high throughputs [15,16]. The second consists of hybrid indexing methods that synthesize the advantages of different indexes to compensate for the existing methods [17]. The hybrid index methods are gradually trending and have become commonly used in distributed file systems and the NoSQL (Not Only Structured Query Language) database, which maximizes I/O via horizontal extension [18]. Hybrid indexing methods can be divided into two categories: general and professional. General indexing methods are typically used in data warehouses, such as Hadoop-GIS [19] and SpatialHadoop [20], which are optimized for high scalability and stability. Professional indexing methods are usually adopted in solutions for customized applications, such as MD-HBase [21] and TrajStore [22], which are optimized for high-insertion rates and real-time querying requirements. Unfortunately, because the distributed file systems and NoSQL database are not natively optimized for spatio-temporal data, the capabilities of the specifically designed spatio-temporal indexes are uneven, and some of them do not provide support for the temporal constraints of spatial data, such as Hadoop-GIS and SpatialHadoop [23].

The indexes in the second group aim for the real-time computation and analysis of multi-modal scene data. Generally, with the increase in main-memory capacity, researchers tend to design internal and external collaborative index structures to store the index and hotspot data in the main memory to achieve faster processing speeds [16]. Indexes in this group usually adopt the global-local hybrid pattern to achieve high efficiency during parallel computing. The global part of this pattern consists of coarse granularity, and algorithms with low maintenance costs are often used to realize the fast division of data or tasks. The local part of this pattern consists of fine granularity, which is responsible for the accurate indexing and management of divided data [24]. The difference between various global-local hybrid pattern indexing methods is the design of the global constituent. Some employ grid or Quad-tree methods as their spatial partitioning technique [25], some adopt the B⁺-tree method as the time interval partitioning technique [26], some use hashing as the random partitioning algorithm [27], and some introduce space-filling curves to convert spatio-temporal information into strings [14,28]. For example, the UQE-Index is a typical global-local hybrid index, based on a key-value store that uses B⁺-tree as the global index to manage time intervals and can support high-insert throughput

and provide an efficient multi-dimensional query simultaneously [26]. However, these techniques are involved in the spatio-temporal patterns of indexes, such as the temporal dimension (primarily) and spatial dimension (secondarily), or vice versa [29,30]. The global-local hybrid structure is not flexible enough to meet the needs of a variety of complex analysis queries.

The goal of the third group of indexes is to efficiently address the multi-dimensional characteristics and complex relationships of objects in cyber-physical-social systems and support human-computer interactive response times. Specifically, with the emergence of interactive devices, such as augmented reality (AR) and virtual reality (VR) devices, anomaly detection, behavior pattern mining, and role judgement have become popular research topics [31]. Meanwhile, a large number of complex spatio-temporal queries have been generated during human-computer interactions; specifically, uncertain explorative queries are often attempted by using indexing methods, which are not known to be a priori; this situation has quickly become a common scenario [32]. In such cases, domain experts need to explore the data before they can figure out what the next query is or even which experiment to perform next [33]. The query patterns of the spatio-temporal indexes mentioned in the previous paragraph are limited to a certain spatio-temporal range or a small number of selected attributes. However, multi-modal dynamic and associated data often contain hundreds of variables and attributes, which change frequently; predefined queries are beneficial only if they can track data patterns or events within a given time limit [34]. In such cases, performing tuning and initialization actions first suffers from the fact that we do not have enough knowledge about the portions of data that are of interest [33]. Trees and graphs are well-known structures that use node-link diagrams to represent data relationships [35]. Trees usually store the topological relationships among spatial objects and use semantic information to perform the node selection and object association [36], and the multi-dimensional characteristics and complex relationships of the scene data are usually stored in the relational model, but joint operations are expensive during analysis and exploration [37,38]. Graph models naturally have the advantage of representing multi-dimensional attributes and relationships among objects, and they have been widely used in the fields of social networks, social security, intelligence recommendations, and biotechnology [39]. In the field of GIS, on the one hand, graph models are often used to build knowledge graphs and support knowledge mining and reasoning [40]. On the other hand, graph models are also used to express the topological structures of objects, such as geographic accessibility and road traffic statuses [41]. Graph-based methods can construct multi-modal subgraphs according to fuzzy query conditions in the interactive process of explorative visualization and solve uncertain query problems by using graph-traversing and pattern-matching algorithms [42].

According to the above challenges, in terms of the indexing and spatio-temporal querying of scene data organization, this paper focuses on a graph-based hybrid spatio-temporal indexing mechanism and its corresponding index construction and optimization algorithm, which are designed for dynamic and associated multi-modal scene data. This paper is organized as follows. Section 2 describes the indexing mechanism for the task-oriented scene data. Section 3 shows the construction and optimization algorithm of the STR-graph index. A prototype system for the STR-graph index method is developed, and an experimental analysis is implemented in Section 4. Finally, the conclusion and discussion are addressed in Section 5.

2. Indexing Mechanism for Task-Oriented Scene Data

The index design should not only understand the characteristics of the multi-modal scene data but also specify the task requirements needed for indexing. To construct an efficient indexing mechanism for multi-modal scene data, this paper starts with the types and characteristics of multi-modal scene data and analyzes the connotations and driving forces behind multi-level visualization tasks. Then, the discrepancy requirements for data scheduling with multi-level visualization tasks are summarized. Finally, a hybrid spatio-temporal index architecture is proposed based on the above discrepancy requirements.

2.1. Multi-Modal Scene Data

Scene data are a type of multi-modal spatio-temporal data that are converged and forced to meet the needs of particular visualization tasks. Scene data are collected from cyber-physical-social systems, which usually refer to the cross fusion and associated mapping of physical spaces, social spaces, and cyber spaces. Scene data in cyber-physical-social systems can be divided into three categories (Figure 1): basic framework data, time series data, and relational data; the detailed classifications and definitions are as follows.

- **Basic framework data:** Basic framework data are the background data of a scene and have spatial locations and shapes that are relatively stable and large-scale and slowly updated characteristics, such as those observed in digital elevation models (DEMs), digital orthophoto maps (DOMs), texture images, 3D models, humans, sensors, symbols and text. This type of data is mainly used for view-only visualization tasks.
- **Time series data:** Time series data are dynamic and have significant changes in the position or state value with time. These data have the characteristics of strong sequences, large scales, and fast updating, such as those observed in human flow, car flow, air flow and water flow data. This kind of scene data can directly display or visualize the results of feature extractions and is mainly for analytical visualization tasks.
- **Relational data:** Relational data are widely used in scene data studies and represent the featured associations of spatio-temporal objects, processes, and events in terms of time, location, theme, and semantics. These data have the characteristics of high dimensions and dynamic and complex interweaving (e.g., social networks, Internet of Things, road networks, and other network data). These data are commonly used for mining and reasoning studies, which mainly involve explorative visualization tasks.

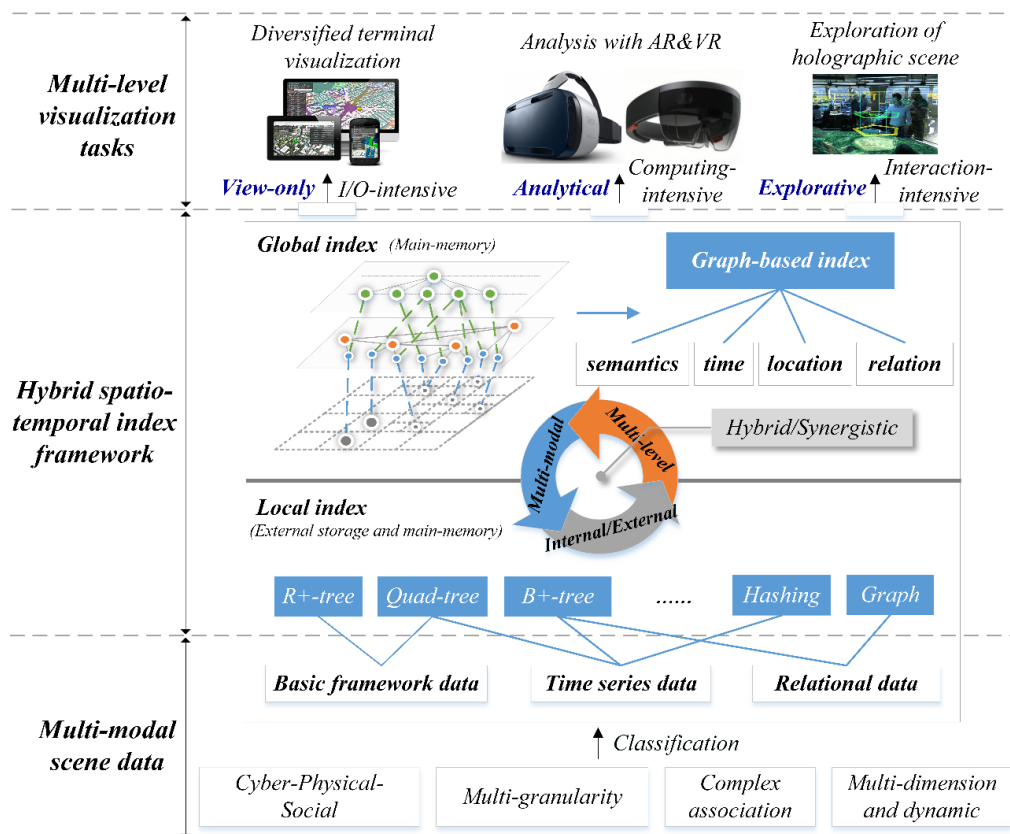


Figure 1. Architecture of the hybrid spatio-temporal index (I/O: Input/Output, AR&VR: Augmented Reality and Virtual Reality).

2.2. Multi-Level Visualization Tasks

According to the different driving forces behind visualization tasks, this paper divided visualization tasks into three levels (Figure 1): view-only, analytical, and explorative, which are defined as follows.

- View-only: these tasks are driven by data and are mainly concerned with the efficient description and transmission of multi-modal spatio-temporal data (e.g., self-adaptive suitability for the visualization and expression of a scene, from discrete to continuous, from macro to micro, and from static to dynamic).
- Analytical: these tasks are driven by data and models and are mainly concerned with sufficiently expressing the implicit information and knowledge of spatio-temporal data obtained via real-time calculations and analysis and highlighting the characteristics and associated relations in the information (knowledge) of an augmented scene (e.g., correlation analysis, dynamic simulation, evolution and prediction).
- Explorative: these tasks are driven by data, models, and interactions and are based on semantic visual variables and new human-computer interfaces. These tasks are characterized by rich human-computer interactive interfaces and fewer pre-conditions, which are accomplished through scene enhancement techniques, such as deformation, highlighting, and simplification, to achieve the goal of combining scene data, human brains, and machine intelligence to support knowledge discovery, hypothesis validation, decision making, and deep association analyses.

Obvious differences exist in the organization and management requirements for multi-modal scene data at different visualization task levels. The view-only visualization task focuses on data parallelism to maximize the I/O and is a typical I/O-intensive task. Therefore, it is necessary to study a local index method that is based on external storage to support the classification indexing and efficient retrieval of display-oriented scene data. Analytical visualization tasks focus on the real-time computation and analysis of the Internet of Things and sensor data. These tasks are a type of typical I/O-intensive and computing-intensive task. Therefore, it is necessary to study a synergistic spatio-temporal indexing method based on the main-memory and external storage to support the real-time access and dynamic scheduling of analysis-oriented scene data. Explorative visualization tasks focus on the use of human-computer interaction visualization techniques to explore the deep features, relationships, and knowledge of multi-modal scene data. These tasks are typical computing-intensive and interaction-intensive tasks. Therefore, it is necessary to study global indexing techniques that support high-dimensional dynamic features and interrelated associations for the fast retrieval, analysis, and support of interactive hypotheses and knowledge discovery.

2.3. Architecture of the Hybrid Spatio-Temporal Index

Considering the features of multi-modal scene data and the driving forces behind multi-level visualization tasks in a scene, this paper proposes a multi-modal and multi-level hybrid spatio-temporal index that contains two parts: a global index and a local index (Figure 1). The global index is a graph-based index that expresses the time, location, semantics, and relation between multi-modal scene data objects and features by using a graph model to quickly query the scene data with spatio-temporal relation information and support explorative interaction reasoning and assumptions. This index is mainly based on the main-memory and supports the fast retrieval of associations and data objects. The local index consists of many fine-grained indexes, including R⁺-tree, Quad-tree, B⁺-tree, hashing, and graph. These indexes exist in the external storage and main-memory, providing efficient data scheduling for I/O-intensive and computing-intensive tasks.

3. STR-Graph Index Algorithm

This section describes the concept and structure of the proposed index method, named the STR-graph, which adopts the hybrid spatio-temporal indexing mechanism discussed in the previous

section, to achieve adaptive aggregation of time series data and associated data for visualization tasks. The graph-based index is a core part of the STR-graph index algorithm, which is introduced in Section 3.1. Section 3.2 introduces the whole framework of the STR-graph index. Finally, the optimization and operation algorithms of the STR-graph index are mentioned in Section 3.3.

3.1. Introduction of the Graph-Based Index

The graph-based index uses nodes, edges, and properties to intuitively express heterogeneous multi-modal spatio-temporal entities and their time, location, relation, and semantic aspects; the framework of the graph-based index is shown in Figure 2. The graph-based index includes time nodes, spatial nodes, and entity nodes. The time nodes can be used to represent the time points and intervals of different granularities; the spatial nodes can be used to represent the spatial ranges of different granularities, and the entity nodes consist of object nodes, feature nodes, and data nodes. The feature nodes are generated from the attributes, events, and behaviors of the object and can be adaptively loaded according to different visualization task topics; the data nodes represent the data generated or associated with the object. Therefore, the graph can be divided into three subgraphs: a time subgraph composed of time nodes; an entity subgraph composed of object nodes, feature nodes, and data nodes; and a spatial subgraph obtained by spatial division. There are three main relationship sources in the graph-based index. The first category is obtained directly from the original input data, including the known feature associations between the objects and extracted spatio-temporal feature associations, as well as the time nodes and spatial nodes. The second category is obtained from the results of the analytical visualization tasks, which are used to update the graph-based index. The third category consists of a temporary relationship that is established through human-computer interactions during the explorative visualization process, which is used for reasoning and assumptions. The relations in the graph-based index can be divided into six categories: spatial, temporal, social, cyber, attributive, and causal, which can be used to express the complex relationships among multi-modal objects in cyber-physical-social systems. Relations can be extended on the basis of these six categories, and Figure 3 shows some typical examples.

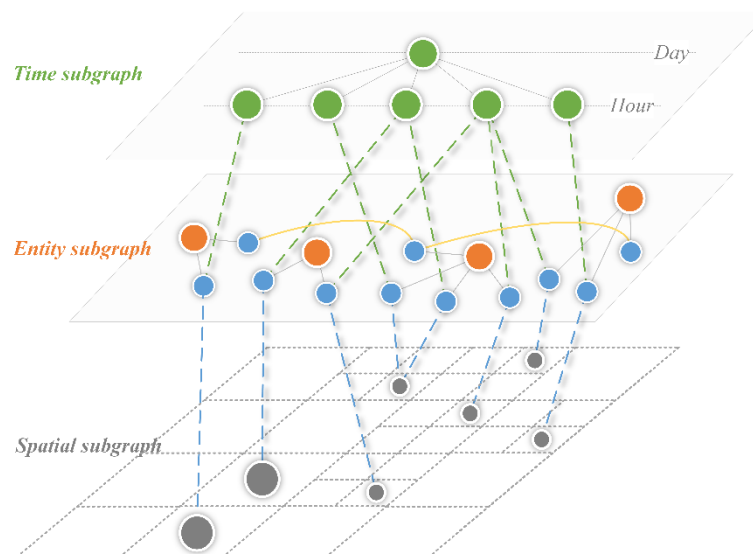


Figure 2. Framework of the graph-based index.

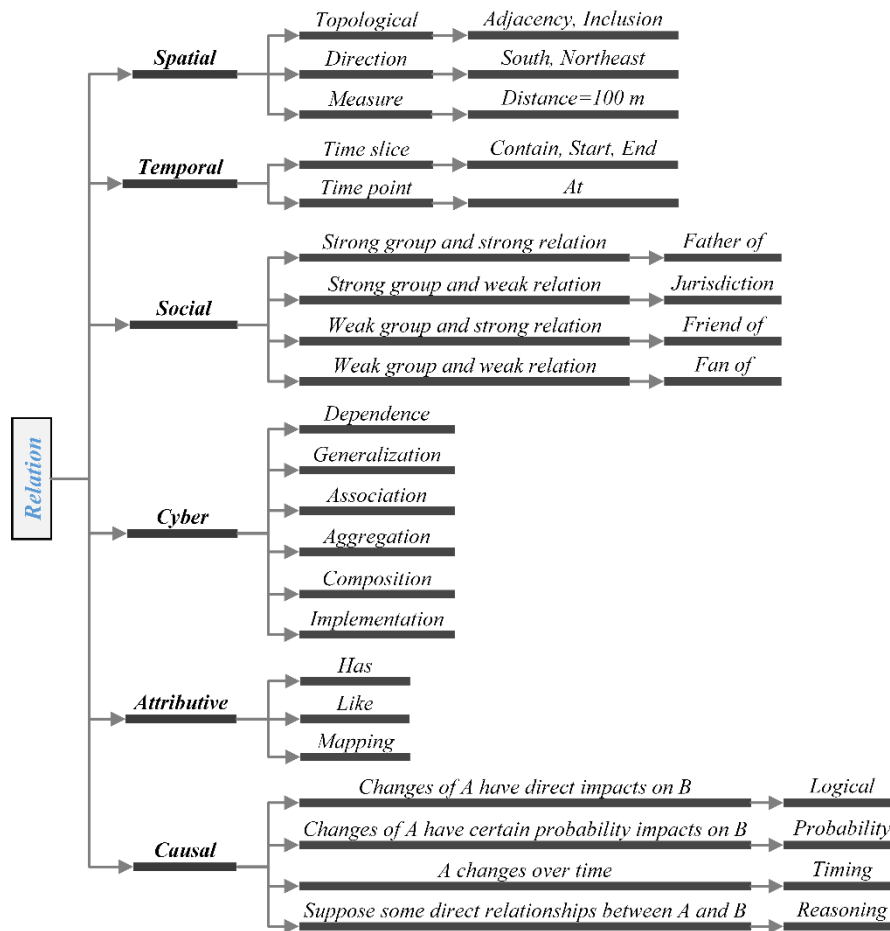


Figure 3. The classification of relations and some typical examples.

The nodes and edges in the graph-based index have the following characteristics. The nodes have labels and attributes, and the edges are directed and can be used to express directed and undirected relations. The relationships among objects are related through feature nodes. The construction of the graph-based index is divided into three steps. The first step is to generate nodes of the graph-based index and includes creating the object, feature, and data nodes according to the input multi-modal scene data. The second step is to compute and generate the corresponding temporal and spatial nodes according to their spatio-temporal information (if none exists). The third step involves creating edges between time nodes, spatial nodes, object nodes, feature nodes, and data nodes based on their spatio-temporal information and known relations. For example, O_2 represents an object that will be synchronized via the graph index; it has a feature node, S , which represents the social relationships and a data node, P , which represents a continuous trajectory for a certain period of time. At the same time, it is known from the input social relationships that O_2 is a fan of O_1 . The construction process and final results of the graph-based index are shown in Figure 4. The delete operation performs the different deleting strategies based on the type of the node, such as the object node, first deleting all the feature nodes, data nodes and their relationships, and then deleting the object node and their relationships.

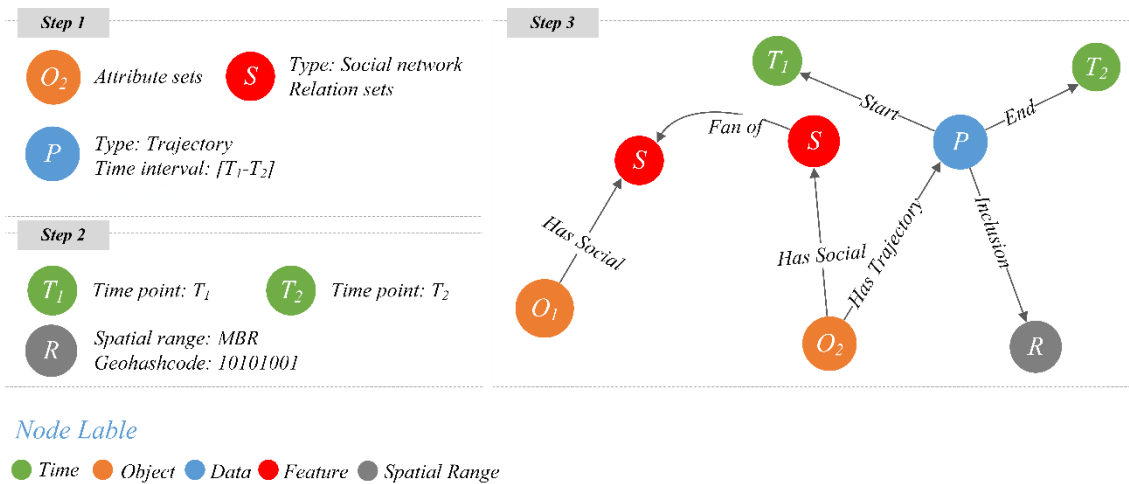


Figure 4. An example of construction of the graph-based index.

The time subgraph and spatial subgraph constitute the spatio-temporal framework of the graph-based index, which allows for spatio-temporal queries. To accelerate spatio-temporal queries, the time nodes are stored as (id, timestamp, type), where id represents a globally unique identifier, timestamp represents a Unix timestamp, and type represents different temporal precisions (e.g., year, day, and hour). The spatial nodes are stored as (id, geostring), where id represents a globally unique identifier and geostring represents encoding of the spatial range. This paper adopts a 52-bit geohash algorithm that transforms longitude and latitude coordinates into unique geocodes to support spatial queries. The geocodes are calculated by using a planar Quad-tree, which divides the earth into $(2^{26}) \times (2^{26})$ grids when the length of the geostring is 52. The Quad-tree is only utilized during the process of computing and not in the actual storage process. For example, a geohash string with ten bits at a given point, with a longitude and latitude of 104.07 and 30.54, respectively, is equal to 1110010011, with a border length of approximately 1,251,296 m; the generation process for the geostring is shown in Figure 5. When the length of the geostring is 52, the border length of the geographic unit is approximately 0.6 m. For a spatial object, the minimum bounding rectangle (MBR) is calculated first. Then, two diagonal points of the above-mentioned MBR are selected and their geostring values are calculated individually with the length L set to 52 as its initial value. If the values are the same, which indicates that the spatial object is completely contained in the correspondent grid, then the geostring is selected to represent the spatial object. Otherwise, L-2 is assigned to L and the calculating process is repeated until an identical geostring value is attained.

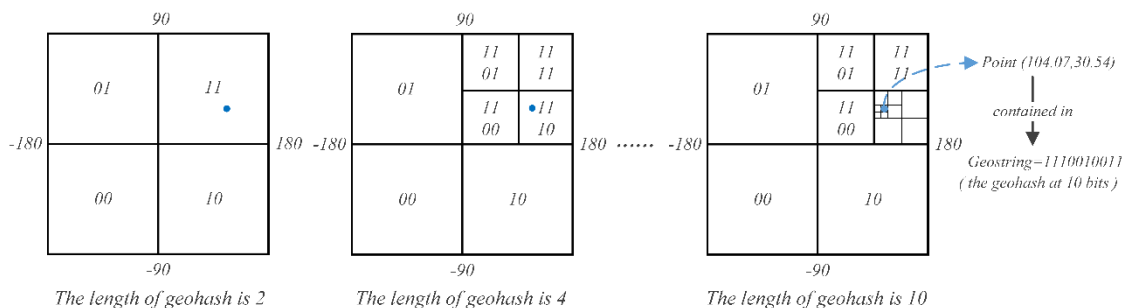


Figure 5. The geostring generation process based on a planar Quad-tree, taking point (104.07, 30.54) as an example.

3.2. Framework of the STR-Graph Index

The efficiency of the STR-graph index is dependent on the cooperation of the global and local indexes. The whole index framework and scene data flow of the STR-graph index are shown in

Figure 6. When the multi-modal scene data enter the system, they are first cached in the main-memory in different HashMaps according to the scene data type. For example, the basic framework data, such as buildings, sensors, and users, are stored as (Key (uid_A), Map (field, property)), and the time series data, such as user trajectories and sensors, are stored as (Key (uid_A), Map (timestamp, value)); the relational data between objects A and B are stored as (Key (uid_A, Map (uid_B, type))). Then, according to the pre-set time interval threshold for each scene data type, the cached scene data are updated in batches into the graph-based index synchronously. After the update is completed, the time series data remain cached in the memory, but the storage key is replaced by the key of the data node. Through the connection between the time node and spatial node, a coarse-grained spatio-temporal index is formed to realize the position of the spatio-temporal data. To improve the efficiency of time series data processing and solve computing-intensive problems, the STR-graph caches the time series data over a period of time, and the data used for the analytical visualization tasks in the memory. In general, the scene data associated with the data nodes in the graph-based index are cached in the memory and dynamically loaded and removed with scene tasks.

The graph-based index server as a global index, which determines the dynamic updating and maintenance of the entire index framework. On the one hand, it simultaneously calculates and updates the nodes and edges of the graph-based index according to the scene data in the memory cache; on the other hand, it decides how to flush the historical data from the memory cache to external storage. To maintain the size of the graph-based index at a reasonable scale, this paper uses a time-based node elimination mechanism to delete the data nodes connected to historical time node t_0 . At the same time, the cached spatio-temporal data corresponding to these data nodes are set up with a fine-grained spatial index according to different data types. Then, the historical data are backed up to the external storage in key/value pairs, where a key is a combination of time nodes and data nodes that is conducive to the rapid loading of historical data during the global display and analysis of a scene. Finally, feature nodes and corresponding relations in the graph-based index are backed up in an external relational database using B⁺-tree to index the fields of interest.

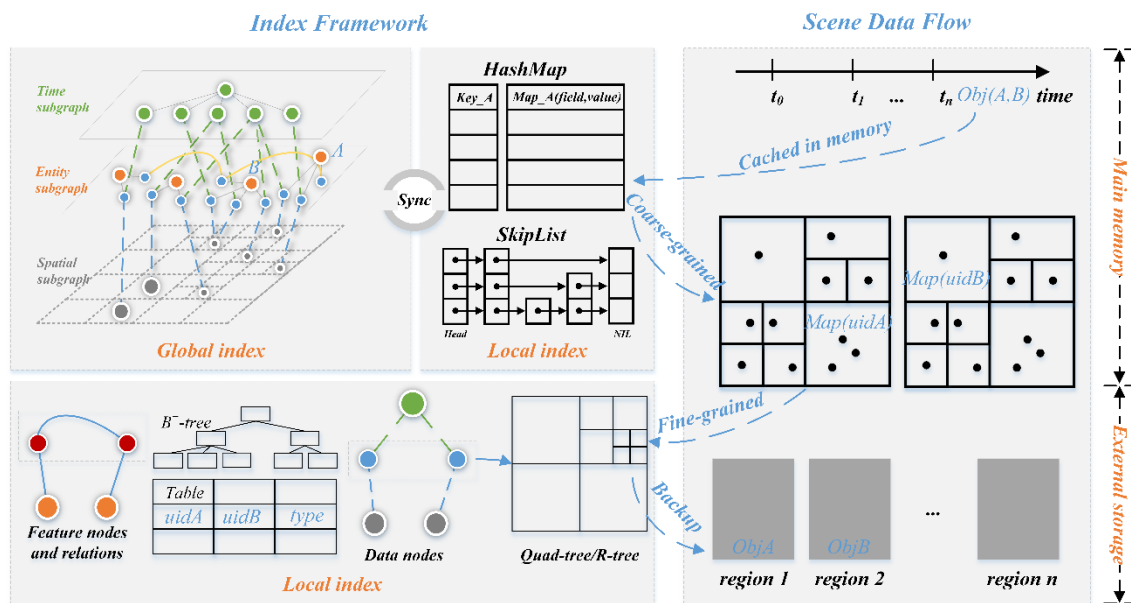


Figure 6. Framework and scene data flow of the spatio-temporal relation graph (STR-graph) index.

3.3. Optimization of the STR-Graph

With the continuous growth of multi-modal scene data, time nodes and spatial nodes will experience a large number of connections with the nodes throughout the entity subgraph, which results in the problem of super node. Given the index-free adjacency features of a graph, each node

server as a “mini index” of its adjacent elements; thus, as the index grows in size, the complexity of traversing the whole graph is equal to $O(n)$, where n represents the number of nodes. However, it is very time consuming to modify or delete an edge of a super node without using indexes. To solve the above problems, this paper adopts a sorted set implemented by HashMap and SkipList to sort the number of connections between the time nodes and spatial nodes. For a node whose connection number exceeds the threshold, it additionally records its connections in the memory to improve the operation efficiency of the super node for the specified relationship. The geostring or timestamp of the node is used as the key; the id values of the entity nodes and relationships are used as the map’s fields and values, respectively. When the number of connections is reduced to a reasonable range, the additional recorded relationships are deleted.

Pattern matching based on a graph-based index can be used to address complex spatio-temporal relation queries and uncertain queries in analytical and explorative visualization tasks by constructing subgraphs of spatio-temporal relationships. When dealing with basic spatio-temporal range queries, the query mode based on graph pattern matching is somewhat complex and inefficient. In particular, when the graph-based index is too large to completely fit into the memory, spatio-temporal matching needs dynamic scheduling of internal and external storage, which does not meet the requirements of view-only visualization tasks. Therefore, regarding the above problems, time subgraphs and spatial subgraphs are cached in the main-memory, and hashing is used to establish mapping between the timestamp and time node id and the geostring and spatial node id, which enables fast spatio-temporal queries and repeated verification.

4. Prototype System Implementation and Experimental Analysis

4.1. Introduction

The scene data management system service is implemented by Java 1.8, while the graph model is supported by Neo4j Community Edition v3.1.1 (a native graph database). The main-memory cache is based on Redis v3.2 (a version of Windows supported by Microsoft), the underlying data node storage system uses a Docker-based HBase v1.2.6 database, and the feature node and relation storage systems use a PostgreSQL v10 database. The experiments are done in a workstation with a 64-bit Windows 10 operating system that has a CPU with Intel Xeon E5-2650 v3 2.3 GHz (dual processors), 128 G of RAM, a 500 G SSD, and a 2×2 TB hard disk (7200 rpm). Neo4j and Redis are deployed on the SSD, and HBase and PostgreSQL are deployed on the hard disks.

To verify the effectiveness of the proposed indexing method in scene data organization and management, some typical scene applications in a smart city were selected for verification. The real-time data and associated data generated in the edge domain environment were organized and managed using the method proposed in this paper, which supported view-only, analytical, and explorative visualizations. The edge domain is an extension of the cloud center, and can manage near real-time applications at the scene side by the edge computing method. The Prototype system is run by the combination of multiple software and platforms, providing high-performance scene data processing and spatial-temporal relation query functions through web service interfaces.

Some index efficiency comparison experiments were conducted with the target indexing method, named the UQE-Index [26]. As mentioned in the introductory section, the UQE-Index is a typical spatio-temporal hybrid indexing method based on NoSQL storage, which is also global-local designed. The UQE-Index contains three levels. The first two levels are the time interval index and subspace index, which are coarse-grained indexes that are responsible for indexing the current data. The third level is a local, fine-grained index that is used to index the historical data. The goals of the UQE-Index are to address the efficiency of real-time access and management of Internet of Things data with efficient I/O performance and a multi-dimensional query ability simultaneously. Therefore, this method is selected to carry out comparative experiments to illustrate the advantages of the STR-graph.

The experimental data are generated by a simulation programme that mainly includes three types of data: user data, trajectory data, and social network data [37]. The details of the experimental datasets are given in Table 1 and screenshots of the experimental data fragments are shown in Figure 7. This set of data can be used to verify the efficiency of the proposed indexing method for time series and complex associated data in the edge domain scene. On the basis of this set of experimental data, time consumption experiments for index generation and for efficiency of scene data scheduling for view-only, analytical and explorative visualization application cases were separately evaluated. The purpose of the experiment was to detect whether the performance of the index can address the dynamic and associated scene data in real time, and whether the retrieval performance of the index can meet the different requirements of multi-level visualization tasks for data scheduling.

Table 1. Descriptions of the experimental datasets.

Dataset Name	# Users	# Relationships	Time Interval	# Trajectories
OBJ10000	10,000	121,716	(5 July 2018 08:00, 5 July 2018 14:00)	4,000,000
OBJ30000	30,000	517,606		12,000,000
OBJ50000	50,000	884,238		20,000,000
OBJ100000	100,000	1,768,515		40,000,000

User data

```
id public completion_percentage gender region other attributes...(separated by spaces)
1 1 14 1 zilinsky kraj, zilina 2012-05-25 11:20:00.0 2005-04-03 00:00:00.0 26 185 cm, 90 kg it
2 1 62 0 zilinsky kraj, kysucke nove mesto 2012-05-25 23:08:00.0 2007-11-30 00:00:00.0 0 166 cm
16 1 64 1 zilinsky kraj, kysucke nove mesto 2012-05-25 23:19:40.0 2008-05-18 00:00:00.0 23 17
3 0 38 1 bratislavsky kraj, bratislava - karlova ves 2012-05-10 18:05:00.0 2010-05-23 00:00:00.0
4 1 12 0 banskobystriicky kraj, brezno 2011-12-29 12:25:00.0 2011-12-29 00:00:00.0 26 null nu
17 1 47 0 zilinsky kraj, martin 2012-05-25 09:40:00.0 2006-10-21 00:00:00.0 27 162 cm, 60 kg i
5 1 66 1 zilinsky kraj, zilina 2012-05-25 01:10:00.0 2006-05-29 00:00:00.0 26 null som nezame:
```

Trajectory data

```
time id Longitude latitude (separated by semicolons)
2018-07-05 08:00:00;1;116.46506;39.80985
2018-07-05 08:00:00;2;116.42704;39.95681
2018-07-05 08:00:00;3;116.33457;40.00722
2018-07-05 08:00:00;4;116.36075;39.82115
2018-07-05 08:00:00;5;116.43642;40.07201
2018-07-05 08:00:00;6;116.29485;39.83154
2018-07-05 08:00:00;7;116.48665;40.06376
```

Social network data

```
id1 id2 type (separated by spaces)
2 171 friend_of
2 172 friend_of
2 173 friend_of
3 1 friend_of
4 1 friend_of
5 1 friend_of
5 178 friend_of
```

Figure 7. The fragments of user data, trajectory data, and social network data.

4.2. Time Performance of Index Generation

Task-oriented scene data need to be time-critically processed, so index generation performance is a crucial aspect. A comparative experiment with different data quantities is conducted between the STR-graph and UQE-Index. To ensure the reliability of the test results, both time intervals of the two indexes are set to 10 min, the minimum geohash length of the spatial node in the STR-graph is 38, and the threshold of connections of time nodes and spatial nodes is twenty thousand. Figure 8 shows the generation and processing speed curves of the STR-graph and the UQE-Index under different data sizes, where the speed is calculated by dividing the total number of data records by the time required for index calculation and storage.

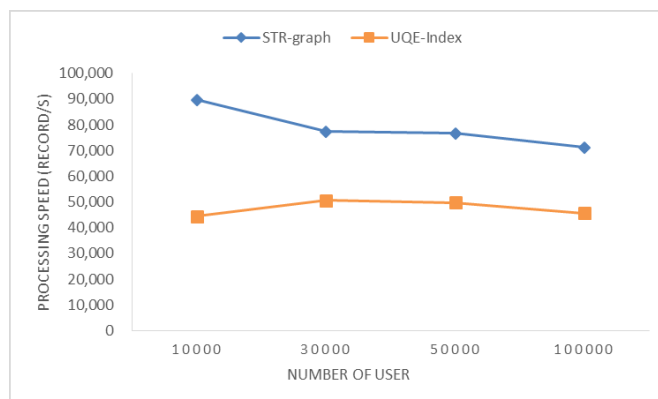


Figure 8. Index processing speed curve of the STR-graph and UQE-Index under different datasets.

In Figure 8, the speed curve shows that the index generation performance of the STR-graph is better than that of the UQE-Index, which can be explained from two aspects. The first aspect is that the STR-graph makes full use of the memory storage medium that stores the global index and the recent scene data in the memory and updates the index in batches through the internal and external storage coordination mechanisms. The second aspect is that the leaf nodes of the tree-based index framework are frequently split and merge during the processing of the scene data, in which the execution of node selection and node splitting is very time consuming. However, the STR-graph organizes the scene data based on the graph model, which greatly reduces the index maintenance overhead in the global and local index parts.

The efficiency curve of the STR-graph shows that the number of connections in time and spatial nodes has a great impact on the index performance. When the test dataset is OBJ10000, the number of data nodes is relatively small, and the number of connections of the time node and spatial node is within the threshold. However, as the number of users increases, the connection number exceeds the threshold, resulting in performance degradation. Fortunately, because the data node elimination mechanism and the super node optimization mechanism were adopted, the overall performance of the index remained stable.

4.3. Spatio-Temporal Relation Query Evaluation

To verify whether the index can achieve adaptive aggregation of task-oriented multi-modal scene data, we designed three scene data retrieval experiments corresponding to the three levels of visualization tasks. In the experiment, the test dataset is OBJ50000, and the parameters of the time node and spatial node of the index are the same as experiment 4.2. The experimental result is the average of multiple test results.

A view-only visualization task is a typical I/O-intensive application, and it is also a basic function for which the spatio-temporal index must be satisfied. Querying in this application scene consists of searching the objects that meet the conditions within a given time interval and spatial range. To evaluate the spatial and temporal data query capabilities between the STR-graph and the UQE-Index, we carry out two spatio-temporal query experiments under different time intervals and spatial ranges. The first experiment is to query spatio-temporal data in a given time interval and in different spatial ranges. The subgraph structure based on pattern matching and the specific temporal and spatial conditions are shown in Figure 9a, and the local index position corresponding to the trajectory data is obtained through pattern matching. Finally, the local index is used for an accurate search. The result of the query time is shown in Figure 10a. The second experiment involves querying spatio-temporal data in a given spatial range and within different time intervals. The subgraph structure based on pattern matching and the specific temporal and spatial conditions is shown in Figure 9b, and the result of the query time is shown in Figure 10b.

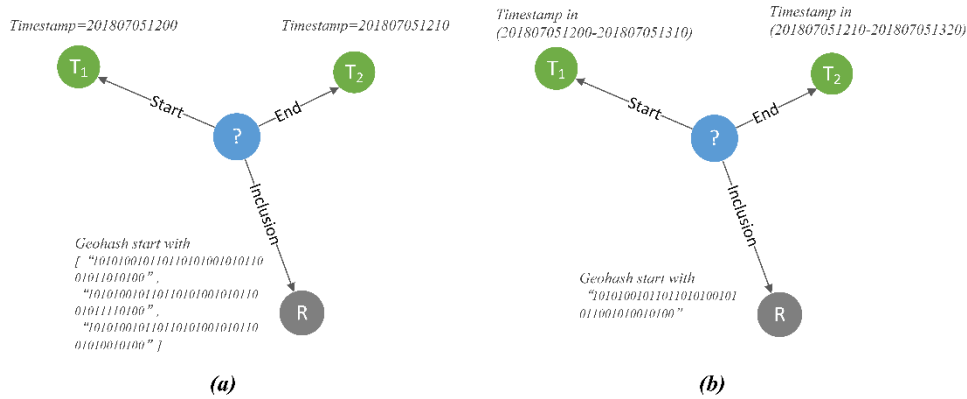


Figure 9. Pattern matching subgraph of the spatio-temporal query: (a) a given time interval with different spatial ranges; (b) a given spatial range with different time intervals.

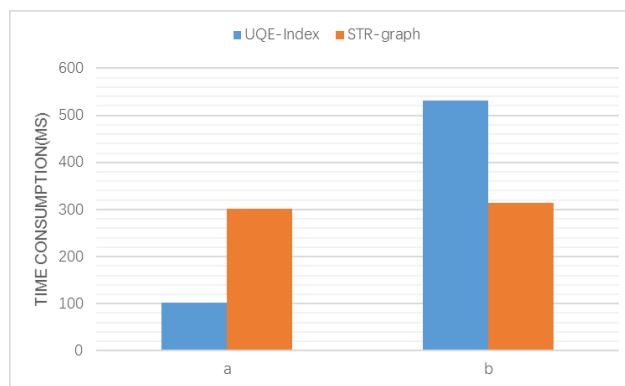


Figure 10. Time consumption results of the spatio-temporal query.

Through the two experiments, the efficiency of the index for different spatio-temporal query patterns is shown in Figure 10. The spatio-temporal design pattern of the UQE-Index is a temporal dimension (primarily) and a spatial dimension (secondarily), which lead to the UQE-Index having advantages when addressing the first experiment, but in regard to the second experiment, the performance of UQE degenerates rapidly because it needs to traverse all possible time intervals to query the data within a given spatial range. However, the performance of the STR-graph is stable because the data nodes are directly connected to the time nodes and spatial nodes in the entity subgraph, and queries are retrieved from time nodes or spatial nodes along the relational edges, reducing unnecessary traversal operations.

An analytical visualization task is a typical computing-intensive application; queries in this application contain not only temporal and spatial information but also multi-dimensional association constraints. To evaluate the ability of a spatio-temporal relation query between the STR-graph and the UQE-Index, we designed a spatio-temporal relation query to determine whether it has the objects, within the four orders of the social network of the object A, of meeting with A in the past hour. The time and spatial ranges in this query are uncertain, so the query should be started with the relationship constraints. The subgraph structure based on pattern matching is shown in Figure 11a, and the results of the experiment are shown in Figure 11b. The STR-graph clearly performs better because the UQE-Index first queries all objects within the four orders through a relational table; then, it queries the trajectories of these objects in the past hour and submits the data to the computing engine to calculate the spatio-temporal distance to determine whether the possibility of meeting exists. However, the STR-graph can quickly retrieve all the relational objects within four orders and then continue to access the data nodes associated with the corresponding time nodes along the relational edges, which avoids unnecessary traversal operations and greatly reduces the redundant data submitted to the computing

engine through pattern matching. Although both the STR-graph and UQE-Index will cache the latest data and computing-oriented hotspot data in memory, the STR-graph constructs the spatio-temporal relationships among objects, trajectories, and features in memory, providing a well-organized index for computing-intensive applications versus packing spatio-temporal data into coarse-grained data blocks.

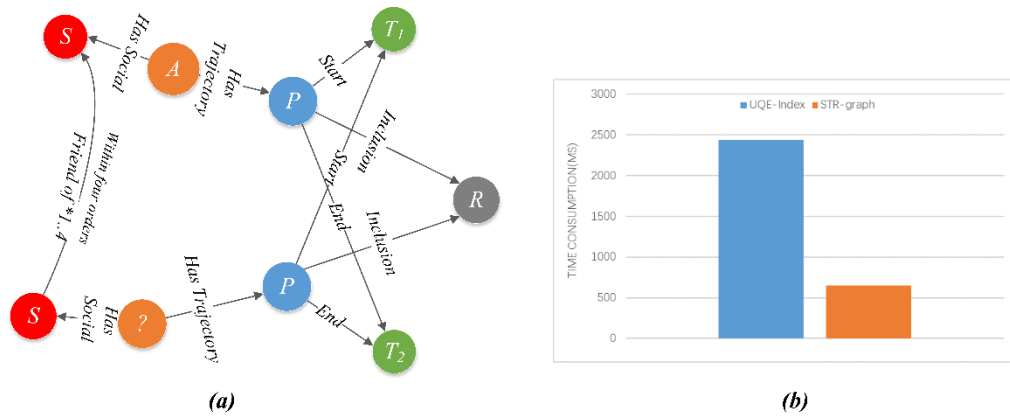


Figure 11. Pattern matching subgraph (a) and time consumption results (b) of the spatio-temporal relation query.

An explorative visualization task is a typical interaction-intensive application; the explorative process involves a large number of human-computer interactions to perform an iterative selection, which relies on the operator's domain knowledge, often accompanied with the hypothesis of some complex relationships. Explorative visualization is the chain of view-only visualization and analytical visualization through human-computer interaction. To demonstrate the role of the STR-graph in the whole process, we designed a case study of explorative visualization based on HoloLens (a mixed reality device), taking community analysis as an example. The detailed operation process of the case study is shown in Figure 12, and the time cost of data scheduling is shown in Table 2. The first step is to build a pattern matching subgraph in HoloLens through the operator view interface based on spatio-temporal query conditions and then check the quality and features of the users in the specified time and spatial range on the display view. Operators can display data sets in a variety of visualizations, such as symbol, chart, 3D, etc., filtering data through knowledge and experience, or re executing spatio-temporal queries to obtain new data sets. The second step is to calculate the shortest path between the remaining objects and return the shortest 1000 results, and then some interference results are eliminated on the display view, for example, the relationship between users will be eliminated in the case of family. The third step is to analyze whether the objects in the remaining results have been out together recently, and then the quality and distribution of the results are checked on the display view via human-computer interaction and domain knowledge. Finally, the hypothesis and association of the results are given to determine the closeness between the users in regard to the whole community network. Because the UQE-Index does not support the computation of the shortest path, no comparison is made. The query time of the STR index is maintained within an acceptable response time in the whole process of explorative visualization, which ensures a good human-computer interactive experience.

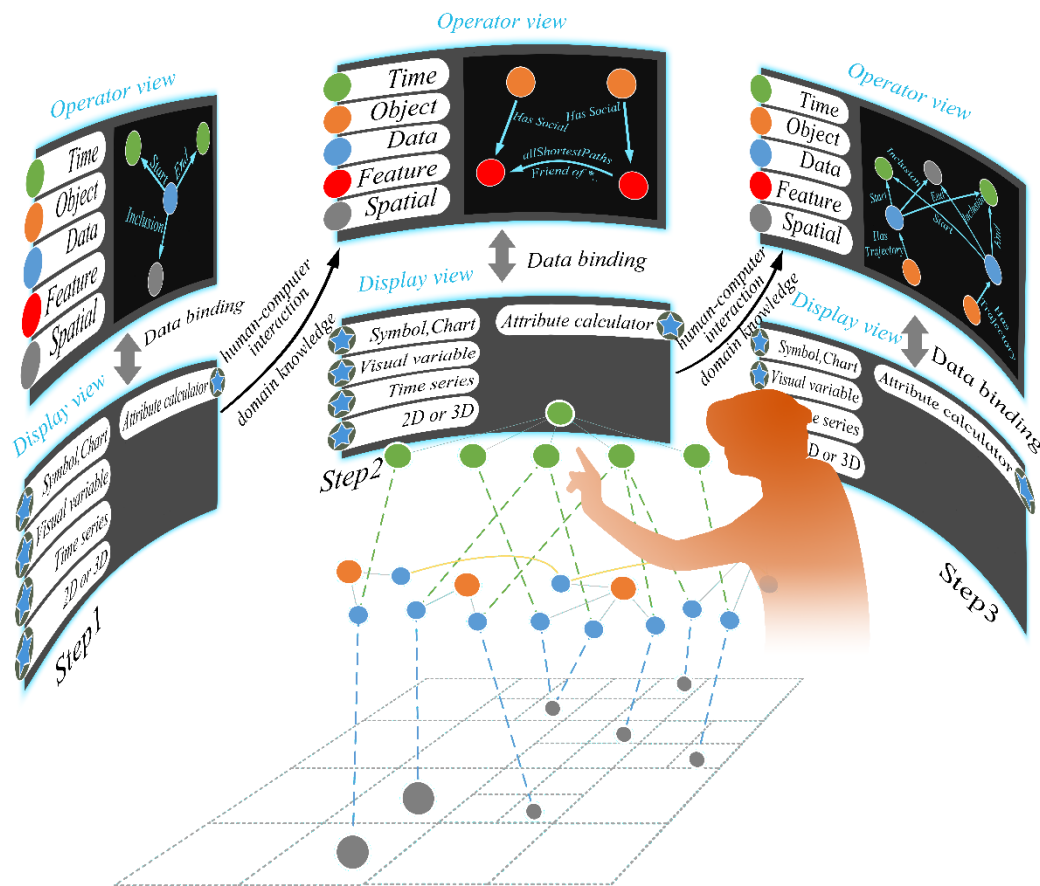


Figure 12. The detailed operation process of explorative visualization based on HoloLens, taking community analysis as a case study.

Table 2. The detailed time consumption of the data query and scheduling in the explorative visualization process.

	Step 1	Step 2	Step 3
Time consumption	318 ms	1362 ms	647 ms

5. Conclusions

Improving the efficiency of task-oriented multi-modal scene data organization is significant, especially for dynamic and associated scene data. In this paper, a new hybrid spatio-temporal index architecture is proposed to meet the requirements of multi-level visualization tasks of the organization and to schedule multi-modal scene data. On the basis of these requirements, an index prototype, named the STR-graph, is built for the dynamic and associated data. This index outperforms the UQE-Index in several aspects, such as the stability of spatial/temporal first query efficiency and uncertain spatial-temporal relation query ability. Therefore, the STR-graph can support high-performance scene data organization and real-time responses. The main contributions of this paper are summarized as follows:

First, a detailed classification of scene data was provided according to the data features in cyber-physical-social systems. The different requirements of multi-level visualization tasks for scene data scheduling were summarized according to the driving forces behind the visualization tasks. An architecture of the global-local hybrid spatio-temporal index, based on the features of multi-modal scene data and the driving forces behind multi-level visualization tasks, was designed for task-oriented scene data organization.

Second, based on the proposed index architecture, the global graph-based index algorithm and the integrated STR-graph index framework were proposed for the dynamic and associated scene data. The operation mechanism of the STR-graph framework was introduced with scene data flow. Moreover, internal and external storage optimization methods and node maintenance strategies were created. The comparative analysis shows that the STR-graph exhibits an efficient spatio-temporal relation query capability, and is fit for I/O-intensive, computing-intensive, and interaction-intensive applications.

Lastly, a prototype system was implemented to support organization and query of scene data. This system has a high-performance throughput capability, and is able to support diversified spatio-temporal relation queries through pattern matching to meet the view-only, analytical, and explorative visualization tasks. In short, this paper provides a new paradigm that will help promote the application of task-oriented multi-modal scene data organization.

Generally, associated and task-oriented real-time scene data organization and management constitute a key technique in the field of the Internet of Things and edge computing. Future studies will improve the basic spatio-temporal query capability of the STR-graph by caching more spatial nodes and time nodes in the main-memory and accelerate the response speed of complex and uncertain spatio-temporal relation queries by pattern matching with multi-dimensional semantic feature constraints. Furthermore, attention will be directed toward the combination of AR and VR devices in the edge domain to fulfil the idea of “think globally, act locally”, supporting efforts for intelligent facilities management and decision-making for smart cities.

Author Contributions: Conceptualization, B.F., Q.Z., M.L. (Mingwei Liu), Y.L. and J.Z.; Data curation, M.L. (Maosu Li); Funding acquisition, Q.Z.; Methodology, B.F.; Project administration, Q.Z.; Resources, H.H. and W.Y.; Software, B.F. and M.L. (Mingwei Liu); Validation, X.F. and J.Z.; Visualization, Y.L.; Writing—original draft, B.F.; Writing—review & editing, Q.Z. and Y.Z.

Funding: This research was funded by the National Natural Science Foundation of China [Grant No. 41471320, 41871321], the National Key Research and Development Program of China [Grant No. 2016YFB0502303], and the Smart Guangzhou Spatio-temporal Information Cloud Platform Construction (Grant No. GZIT2016-A5-147).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shekhar, S.; Gunturi, V.; Evans, M.R.; Yang, K.S. Spatial big-data challenges intersecting mobility and cloud computing. In Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access, Scottsdale, AZ, USA, 20 May 2012; pp. 1–6.
2. Chen, C.L.P.; Zhang, C.Y. Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Inf. Sci.* **2014**, *275*, 314–347. [[CrossRef](#)]
3. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **2013**, *29*, 1645–1660. [[CrossRef](#)]
4. Smirnov, A.; Levashova, T.; Shilov, N.; Sandkuhl, K. Ontology for Cyber-Physical-Social Systems Self-Organisation. In Proceedings of the 16th Conference of Open Innovations Association FRUCT, Oulu, Finland, 27–31 October 2014; pp. 101–107.
5. Rathore, M.M.; Ahmad, A.; Paul, A.; Rho, S. Urban planning and building smart cities based on the Internet of Things using Big Data analytics. *Comput. Netw.* **2016**, *101*, 63–80. [[CrossRef](#)]
6. Fox, A.; Eichelberger, C.; Hughes, J.; Lyon, S. Spatio-temporal indexing in non-relational distributed databases. In Proceedings of the 2013 IEEE International Conference on Big Data, Silicon Valley, CA, USA, 6–9 October 2013; pp. 291–299.
7. Rezgui, A.; Malik, Z.; Xia, J.; Liu, K.; Yang, C. Data-intensive spatial indexing on the clouds. *Procedia Comput. Sci.* **2013**, *18*, 2615–2618. [[CrossRef](#)]
8. Ke, S.; Gong, J.; Li, S.; Zhu, Q.; Liu, X.; Zhang, Y. A hybrid spatio-temporal data indexing method for trajectory databases. *Sensors* **2014**, *14*, 12990–13005. [[CrossRef](#)] [[PubMed](#)]
9. Gan, L.; Li, J.; Jing, N. Hybrid organization and visualization of the DSM combined with 3D building model. In Proceedings of the International Conference on Image, Vision and Computing, Chengdu, China, 2–4 June 2017; pp. 566–571.

10. Gao, Q.; Ji, M.; Pang, L.; Wang, X.T.; Li, J.; Wang, M.Q. Research of Large-Scale Terrain Data Organization Method in Virtual Reality. In Proceedings of the International Symposium on Computational Intelligence and Design, Hangzhou, China, 10–11 December 2016; pp. 108–111.
11. Kehrer, J.; Hauser, H. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 495–513. [[CrossRef](#)] [[PubMed](#)]
12. Lahat, D.; Adali, T.; Jutten, C. Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects. *Proc. IEEE* **2015**, *103*, 1449–1477. [[CrossRef](#)]
13. Liu, M.; Zhu, J.; Zhu, Q.; Qi, H.; Yin, L.; Zhang, X.; Feng, B.; He, H.; Yang, W.; Chen, L. Optimization of simulation and visualization analysis of dam-failure flood disaster for diverse computing systems. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 1891–1906. [[CrossRef](#)]
14. Van Le, H.; Takasu, A. A scalable spatio-temporal data storage for intelligent transportation systems based on HBase. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC), Las Palmas, Spain, 15–18 September 2015; pp. 2733–2738.
15. He, Z.; Wu, C.; Liu, G.; Zheng, Z.; Tian, Y. Decomposition tree: A spatio-temporal indexing method for movement big data. *Clust. Comput.* **2015**, *18*, 1481–1492. [[CrossRef](#)]
16. Ray, S.; Blanco, R.; Goel, A.K. High performance location-based services in a main-memory database. *GeoInformatica* **2016**, *21*, 1–30. [[CrossRef](#)]
17. Gong, J.; Zhu, Q.; Zhong, R.; Zhang, Y.; Xie, X. An Efficient Point Cloud Management Method Based on a 3D R-Tree. *Photogramm. Eng. Remote Sens.* **2015**, *74*, 373–381. [[CrossRef](#)]
18. Wu, C.; Zhu, Q.; Zhang, Y.; Du, Z.; Ye, X.; Qin, H.; Zhou, Y. A NoSQL–SQL Hybrid Organization and Management Approach for Real-Time Geospatial Data: A Case Study of Public Security Video Surveillance. *ISPRS Int. J. Geo-Inf.* **2017**, *6*, 21. [[CrossRef](#)]
19. Aji, A.; Wang, F.; Vo, H.; Lee, R.; Liu, Q.; Zhang, X.; Saltz, J. Hadoop gis: A high performance spatial data warehousing system over mapreduce. *Proc. VLDB Endow.* **2013**, *6*, 1009–1020. [[CrossRef](#)]
20. Eldawy, A.; Mokbel, M.F. SpatialHadoop: A MapReduce framework for spatial data. In Proceedings of the 2015 IEEE 31st International Conference on Data Engineering (ICDE), Seoul, Korea, 13–17 April 2015; pp. 1352–1363.
21. Nishimura, S.; Das, S.; Agrawal, D.; El Abbadi, A. Md-hbase: A scalable multi-dimensional data infrastructure for location aware services. In Proceedings of the 2011 12th IEEE International Conference on Mobile Data Management (MDM), Lulea, Sweden, 6–9 June 2011; pp. 7–16.
22. Cudre-Mauroux, P.; Wu, E.; Madden, S. Trajstore: An adaptive storage system for very large trajectory data sets. In Proceedings of the 2010 IEEE 26th International Conference on Data Engineering (ICDE), Long Beach, CA, USA, 1–6 March 2010; pp. 109–120.
23. Bao, J.; Li, R.; Yi, X.; Zheng, Y. Managing massive trajectories on the cloud. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Burlingame, CA, USA, 31 October–3 November 2016; p. 41.
24. Zhou, X.; Li, H.; Zhang, X.; Wang, S.; Ma, Y.; Liu, K.; Zhu, M.; Huang, M. ABR-Tree: An Efficient Distributed Multidimensional Indexing Approach for Massive Data. In Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing, Zhangjiajie, China, 18–20 November 2015; pp. 781–790. [[CrossRef](#)]
25. Eldawy, A.; Alarabi, L.; Mokbel, M.F. Spatial partitioning techniques in SpatialHadoop. *Proc. VLDB Endow.* **2015**, *8*, 1602–1605. [[CrossRef](#)]
26. Ma, Y.; Rao, J.; Hu, W.; Meng, X.; Han, X.; Zhang, Y.; Chai, Y.; Liu, C. An efficient index for massive IOT data in cloud environment. In Proceedings of the 21st ACM international conference on Information and Knowledge Management, Maui, HI, USA, 29 October–2 November 2012; pp. 2129–2133.
27. Li, S.; Hu, S.; Ganti, R.K.; Srivatsa, M.; Abdelzaher, T.F. Pyro: A Spatial-Temporal Big-Data Storage System. In Proceedings of the USENIX Annual Technical Conference, Santa Clara, CA, USA, 8–10 July 2015; pp. 97–109.
28. Hulbert, A.; Kunicki, T.; Hughes, J.N.; Fox, A.D.; Eichelberger, C.N. An experimental study of big spatial data systems. In Proceedings of the 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, USA, 5–8 December 2016; pp. 2664–2671.
29. Mokbel, M.F.; Ghanem, T.M.; Aref, W.G. Spatio-temporal access methods. *IEEE Data Eng. Bull.* **2003**, *26*, 40–49.

30. Nguyen-Dinh, L.-V.; Aref, W.G.; Mokbel, M. Spatio-temporal access methods: Part 2 (2003–2010). *IEEE Data Eng. Bull.* **2010**, *33*, 46–55.
31. Olshannikova, E.; Ometov, A.; Koucheryavy, Y.; Olsson, T. Visualizing Big Data with augmented and virtual reality: Challenges and research agenda. *J. Big Data* **2015**, *2*, 22. [[CrossRef](#)]
32. Zheng, Y.; Wu, W.; Chen, Y.; Qu, H.; Ni, L.M. Visual Analytics in Urban Computing: An Overview. *IEEE Trans. Big Data* **2016**, *2*, 276–296. [[CrossRef](#)]
33. Zoumpatianos, K.; Idreos, S.; Palpanas, T. Indexing for interactive exploration of big data series. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, Snowbird, UT, USA, 22–27 June 2014; pp. 1555–1566.
34. Ferreira, N.; Poco, J.; Vo, H.T.; Freire, J.; Silva, C.T. Visual exploration of big spatio-temporal urban data: A study of New York City taxi trips. *IEEE Trans. Vis. Comput. Graph.* **2013**, *19*, 2149–2158. [[CrossRef](#)] [[PubMed](#)]
35. Wang, C.; Tao, J. Graphs in Scientific Visualization: A Survey. *Comput. Graph. Forum* **2017**, *36*, 263–287. [[CrossRef](#)]
36. Duan, H.; Meng, L.; Huang, C.; Li, Y.; Li, J. A method for geo semantic spatial index on SPARQL query. *Acta Geod. Cartogr. Sin.* **2014**, *43*, 193–199.
37. Płuciennik, T.; Płuciennik-Psota, E. *Using Graph Database in Spatial Data Generation*; Springer International Publishing: Cham, Switzerland, 2014; pp. 643–650.
38. Vukotic, A.; Watt, N.; Abedrabbo, T.; Fox, D.; Partner, J. *Neo4j in Action*; Manning Publications Co.: Greenwich, CT, USA, 2014.
39. Bezerianos, A.; Chevalier, F.; Dragicevic, P.; Elmqvist, N.; Fekete, J.D. GraphDice: A System for Exploring Multivariate Social Networks. *Comput. Graph. Forum* **2010**, *29*, 863–872. [[CrossRef](#)]
40. Jayaram, N.; Khan, A.; Li, C.; Yan, X.; Elmasri, R. Querying Knowledge Graphs by Example Entity Tuples. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 2797–2811. [[CrossRef](#)]
41. Huang, X.; Zhao, Y.; Ma, C.; Yang, J.; Ye, X.; Zhang, C. TrajGraph: A Graph-Based Visual Analytics Approach to Studying Urban Network Centralities Using Taxi Trajectory Data. *IEEE Trans. Vis. Comput. Graph.* **2015**, *22*, 160–169. [[CrossRef](#)] [[PubMed](#)]
42. Holzschuher, F. Performance of graph query languages: comparison of cypher, gremlin and native access in Neo4j. In Proceedings of the Joint EDBT/ICDT 2013 Workshop GraphQ, Genoa, Italy, 18–22 March 2013; pp. 195–204.



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).