Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Relative space-based GIS data model to analyze the group dynamics of moving objects

Mingxiang Feng\textsuperscript{a,b}, Shih-Lung Shaw\textsuperscript{c,*}, Zhixiang Fang\textsuperscript{a,b,*}, Hao Cheng\textsuperscript{d}

\textsuperscript{a} State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, 129 Luoyu Road, Wuhan 430079, Hubei, PR China
\textsuperscript{b} Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan 430079, PR China
\textsuperscript{c} Department of Geography, University of Tennessee, Knoxville, TN, USA
\textsuperscript{d} School of Computer Science and Technology, Wuhan University of Technology, 122 Luoshi Road, Wuhan, Hubei, PR China

ARTICLE INFO

Keywords:
GIS data model
Relative space
Moving objects
Spatiotemporal analysis
Human dynamics

ABSTRACT

The relative motion of moving objects is an essential research topic in geographical information science (GIScience), which supports the innovation of geodatabases, spatial indexing, and geospatial services. This analysis is very popular in the domains of urban governance, transportation engineering, logistics, and geospatial information services for individuals or industries. Importantly, data models of moving objects are one of the most crucial approaches to support the analysis for dynamic relative motion between moving objects, even in the age of big data and cloud computing. Traditional geographic information systems (GIS) usually organize moving objects as point objects in absolute coordinated space. The derivation of relative motions among moving objects is not efficient because of the additional geo-computation of transformation between absolute space and relative space. Therefore, current GISs require an innovative approach to directly store, analyze and interpret the relative relationships of moving objects to support their efficient analysis. This paper proposes a relative space-based GIS data model of moving objects (RSMO) to construct, operate and analyze moving objects’ relationships and introduces two algorithms (relationship querying and relative relationship dynamic pattern matching) to derive and analyze the dynamic relationships of moving objects. Three scenarios (epidemic spreading, tracker finding, and motion-trend derivation of nearby crowds) are implemented to demonstrate the feasibility of the proposed model. The experimental results indicate the execution times of the proposed model are approximately 5–50% those of the absolute GIS method for the same function of these three scenarios. It’s better computational performance of the proposed model when analyzing the relative relationships of moving objects than the absolute methods in a famous commercial GIS software based on this experimental results. The proposed approach fills the gap of traditional GIS and shows promise for relative space-based geo-computation, analysis, and service.

1. Introduction

Moving objects are the most common and important component in a diverse range of phenomena, such as human mobility (Fang et al., 2017; Jiang et al., 2017; AlMuhisen et al., 2018), urban transportation (Liu et al., 2015, Tang et al., 2015; Tu et al., 2017), ship logistics in the ocean (Yu et al., 2017b; Fang et al., 2018) and even animal migrations (Bastille-Rousseau et al., 2017). Many research projects have been driven and improved by moving-object data analysis, such as individual/group behavior analysis, path discovery and behavior prediction. Because of the large amount of moving objects in real applications, these analyses require a powerful GIS data model to store, analyze and interpret physical and contextual information (e.g., absolute topology and relative motion) of moving objects.

Current GIS data models usually record essential information for moving objects within an absolute space. In absolute space, geocoded locations are bound to previously existing geometry and topology relationships among the corresponding points in the space (Meentemeyer, 1989; Couclelis, 1999). Therefore, moving objects are always represented as a series of observations that consist of an id, location and time (Hornsby and Egenhofer, 2002; Spaccapietra et al., 2008), alongside additional information such as activities (Wang and Cheng, 2001; Shaw and Yu, 2009) and semantics (Zhang et al., 2014; Vandecasteele et al., 2014). Furthermore, these models describe their movement processes and interactions alongside basic geographical information (such as land use, POI, and events). In fact, these models require a large
amount of computation to support analysis from the perspectives of individual or groups of moving objects (which could be called relative space). Moreover, several analysis, such as the surrounding dynamics and motion trends of nearby crowds, are critical to moving objects in highly complex and dynamic environments according to the personal requirements of decision-making (e.g., on relax life style, feeling of safety). Current GIS data models must be improved in terms of the geo-computation of these analyses.

Actually, relative space is an instinctual approach to moving objects and is a powerful theoretic framework to represent the surrounding dynamics and motion trends of moving objects in nearby crowds. Traditional relative space is often studied in the research communities of mathematical theories (Einstein, 1921), physics (Veltean, 1994; Ruggiero, 2003), astronomy (Rots et al., 2015) and aerospace science (Sinclair et al., 2015). Currently, very few GIS data models have been built for relative space, which requires additional space-transformation computations to implement this instinctual approach in applications.

In relative space, relative dynamic relationships between moving objects are easy to build independently of whether they can be geo-coded by coordinates. Importantly, the analysis of moving objects in relative space could easily follow instinctual requirements. Therefore, the motivation of this paper is to create a relative space-based GIS data model of moving objects and propose some basic GIS operators for analyzing moving objects, which changes the analysis of current absolute space-based GIS models and facilitates the efficient computation of real-time relative relationship dynamics, such as the surrounding dynamics and motion trends of crowds near moving objects. The contributions from this paper are summarized as follows:

- A relative space-based GIS data model of moving objects (RSMO) is introduced to construct, operate and analyze moving objects’ relative relationships.
- A relationship-querying algorithm and relative dynamic pattern-matching algorithm are introduced to analyze the dynamic relationships of moving objects.
- Three scenarios (epidemic spreading, tracker finding, and derivation of the motion trends of nearby crowds) are implemented to demonstrate the feasibility of the proposed model, which also shows better performance compared to absolute methods.

This paper is structured as follows. Related work is discussed in Section 2. Section 3 proposes the modeling process, structure and operation of this model. Section 4 illustrates the implementation of RSMO and the two algorithms. Experiments with three case studies and performance testing are reported in Section 5. Section 6 discusses the proposed model. Finally, conclusions are presented in Section 7.

2. Related work

Moving objects are a very popular representation requirement in current GIS applications because of the large volume of their tracked trajectories, e.g., cars in cities, vessels at sea, and even people around the world. Therefore, many research projects have been conducted to store, manage and query moving-object data in the communities of computer science and geographical information science. In computer science, several data structures have been defined to support the storage and querying of moving objects. The first structure is the storage unit in DBMSs (database management systems). Güting and his group abstracted moving objects into moving points and moving regions and developed a discrete model in a DBMS to store, manage and query moving objects’ data (Forlizzi et al., 2000; Güting et al., 2000, 2004; Güting and Schneider, 2005). Based on abstracted data types, moving objects could be managed and queried by using SQL. These works provided a solid foundation to support research on moving objects. The second is tree structures. Moving objects were usually abstracted as points (Ranu et al., 2015), single continuous polynomials (Ni and Ravishankar, 2007), or non-regulated sequences of roads in transportation networks (Sandu Popa et al., 2011). Then, these objects were indexed by the tree’s derivatives, for example, B + -trees (Sandu Popa et al., 2011), R-trees (Yue et al., 2018), 3D R-trees (Xu et al., 2018), FN (fixed network) R-trees (Zheng et al., 2011), PA-trees (Ni and Ravishankar, 2007), and grid trees (Yan et al., 2015). These tree-based indices could facilitate efficient queries on moving objects. The third structure is moving object-oriented databases. To satisfy the demand of faster processing with big data, PostgreSQL (Ding et al., 2015), MongoDB (Liu et al., 2013), MonetDB (Boncz et al., 2005), and Cassandra (Hospital et al., 2015) were developed to store and analyze large volumes of moving objects’ information.

In the community of geographical information science, traditional GIS data models, such as field-based models (e.g., raster-based models) (Cova and Goodchild, 2002) and feature models (e.g., object- and vector-based models) (Tang et al., 1996), usually depended on absolute coordinates in their reference frames to describe spatial object’s motion or relationships. These models were often used to represent some key concepts that were related to moving objects, i.e., the place, activity and semantics. The concept of “place” was used to indicate the location of activity, which usually complied with a certain upper-bound distance and lower-bound duration (Kang et al., 2004). Human daily activities were always represented as a sequence of place visits (Do and Gatica-Perez, 2014). Additionally, space time path was a key concept in classical time geography framework that was used to represent and analyze activities and interactions in a hybrid physical–virtual space (Shaw and Yu, 2009) and social space (Yin and Shaw, 2015). Several basic semantics of moving objects were derived from the trajectories of moving objects, for example, episodes, stops, and moves (Ilarri et al., 2015). Based on these basic semantics, researchers (Parent et al., 2013; Bogorny et al., 2014; Jiang et al., 2015; Wan et al., 2018) built semantic trajectories from semantic point sequences. Semantic trajectories were easy for urban planners and migration or transportation agencies to use to mine spatial-temporal mobility patterns.

Moving object-oriented operators (i.e., range queries, nearest-neighbor queries) were derived to support the analysis of moving objects’ relationships. The first common operator is range queries, which specify a value of moving objects to fall within a lower and upper boundary (Zheng et al., 2011; Zhan et al., 2015; Xu et al., 2018; Yue et al., 2018), e.g., finding all the objects of a specific traveler between 7 am and 9 am. Filtering and refining are two important steps of range queries. Filtering determines candidate locations (such as a node in a tree structure) that contain specified attribute values and overlap the query area in space and time, while refining retrieves well-matched objects under the querying conditions through some advanced filtering techniques, such as statistics-based approaches (Zhan et al., 2015) and time-based partitioning techniques (Yue et al., 2018). The second common operation is nearest-neighbor (NN) queries, which find the nearest neighbors to a given object in a database for a specific time interval. Two similar phases of NN queries include searching for candidate units based on the idea of node pruning with large coverage and determining the well-matched results according to various conditions, such as time limits (Güting et al., 2010), obstacles (Gao et al., 2011), spatial-network structures (Cheema et al., 2012), reverse nearest neighbors (Cheema et al., 2012) and movement uncertainties (Niedermayer et al., 2013). These operators are usually applied by calculating the distance between the coordinates of moving objects to derive relative metrics such as the distance, direction and time. This situation could produce repeated computations and misrepresent the relative relationships of moving objects without a geocoding technique.

A wide spectrum of applications can be conducted for individuals, groups or the public. The first application is the movement-behavior analysis of moving objects, for example, individuals (González et al., 2008; Ando and Suzuki, 2011; Gao et al., 2013; Renso et al., 2013; Song et al., 2014) and groups (Zheng et al., 2014; Li et al., 2013; Gupta et al., 2013; McGuire et al., 2014; Liu et al., 2016). The second application is
path recommendation from historical trajectories (Luo et al., 2013; Dai et al., 2015; Yang et al., 2017; Zheng et al., 2018). The third application is location prediction for tourists (Lee et al., 2016; Yu et al., 2017a), navigators (Li et al., 2016; Besse et al., 2018), and driverless vehicles (Wang et al., 2016). These applications require that the dynamic relationships of moving objects in space over time are inferred. However, very few models can directly organize the dynamic relationships of moving objects. These applications greatly depend on computationally intensive infrastructures.

The studies about relative space mainly appear in the fields of vessels in maritime and airplane in aviation. In maritime, the moving object data model also can be applied for assessing collision risk (Bye and Aalberg, 2018; Fang et al., 2018), predicting vessel behavior (Zisis et al., 2016; Xiao et al., 2017) and planning path (Cummings et al., 2010; Hornauer et al., 2015). In aviation field, motion guidance and control (Yu et al., 2016; Sun et al., 2017; Li and Zhu, 2018; Zhu et al., 2018) for spacecraft rendezvous, position and attitude estimation (Philip and Ananthasayanam, 2003; Qiao et al., 2013) between the chaser and the target satellites were also referred to the applications of moving object data model. These studies aim to control the moving object in real time to avoid risk and complete the established movement in the ocean or aerospace, where no appropriate global reference for them. Those studies focus on the real-time motion control to respond to surrounding moving objects in a high dynamic scenario. These missions need an efficient data model to handle relationships of moving objects as well.

In short, current research on moving objects rarely organizes their dynamic relationships directly because of a lack of relative space-based data models and analytic frameworks in the communities of computer science and geographical information science. This paper attempts to fill this gap by introducing a relative space-based GIS data model to analyze the group dynamics of moving objects, which should reduce the intensive computation that occurs when deriving the relationships of moving objects and provide better analytic performance than traditional absolute coordinate-based GIS data models.

3. Proposed relative space-based GIS data model of moving objects

This section introduces a relative space-based GIS data model of moving objects (RSMO). RSMO is extended from the space-time cube (STC) structure in ArcGIS. An STC is a three-dimensional cube that includes space-time bins in the \( x \) and \( y \) dimensions and time in the \( t \) dimension and represents limited space coverage over a fixed period. The basic idea of RSMO is to directly record moving objects’ relationships and relative dynamics with a three-dimensional space-time cube to facilitate transfer to absolute space by maintaining the current locations of moving objects via additional space-time bins. This data model reduces the calculation of moving objects’ relationships and relative dynamics in traditional GIS data models to provide efficient relative space-based queries between moving objects for real applications with cars, vessels, people, etc. RSMO is designed to support some basic functions such as moving objects’ motion storage and organization, queries based on relative relationships, relative motion-pattern analysis and mining, and transformation between absolute space and relative space. The following section will first introduce RSMO and then the basic functions that depend on RSMO.

3.1. Basic idea of RSMO

Before introducing the proposed RSMO, this section defines some basic concepts at first as follows:

Definition 1: Space-time cube (STC) is a three-dimensional (3D) Euclidean space containing of a 2D geographical space \((x, y)\) plus time \(t\) for visualization and analysis of objects’ behavior and interactions across space and time (Bach et al., 2017).

Definition 2: Space-time bin (STB) is a basic unit of space-time cube, which lies in the fixed position based on their \(x\) and \(y\) dimensions in 2D geographical space and \(t\) dimension in time to represent a limited location with a certain time.

Definition 3: Relative Relationship Bin (RRB) is a derived STB with a substituted reference framework of reference object, target object and time (Fig. 1). It contains a quadruple \(< t, \text{ref}_\text{obj}, \text{tar}_\text{obj}, \text{relationship} \rangle\), which indicates the relationships of corresponding reference object (\(\text{ref}_\text{obj}\)) and target object (\(\text{tar}_\text{obj}\)) during a specific time \(t\).

Definition 4: Relative Relationship Matrix is a matrix of all moving objects’ RRBs in the time \(t\) and within a specific environment.

Definition 5: Relative Relationship Cube is a time series of Relative Relationship Matrices, which is organized in three-dimensional space (two dimensions in space and one dimension in time) according to time.

The proposed RSMO model extends the structural organization of space-time bins in ArcGIS to model the relative relationships among moving objects and facilitate their relative space-based analysis. Here, the motion in the STC is represented by the relationships of moving objects, which is a natural approach to represent the actual cognitive processes of moving objects in an environment.

Fig. 2 illustrates the extended-entity-relationship (EER) diagram of the RSMO model, which shows the elements (entities, relationships and attributes) and the hierarchical relationships among these elements. This model defines six basic entity types, i.e., “Object”, “Relative Location”, “Relationship”, “Relative Relationship Bin”, “Relative Relationship Matrix” and “Relative Relationship Cube”. Here, Object represents the moving object in a real scenario, such as the people in Fig. 2, vehicles/unmanned ground vehicles, unmanned aerial vehicles, vessels, planes, and so on. Relative Location is the position of any Object relative to other Object (Fig. 2), which can be represented by the relative distance and angle. Relationship indicates the mutual spatial or social relationships among Objects, for example, closeness, friendship, colleague
relationships and others. Relative Relationship Bin is a three-dimensional unit in space and time, which indicates the Relationship of two corresponding Objects during a specific time $t$. Relative Relationship Matrix is a matrix of Relative Relationship Bins and represents all the mutual relationships of objects at a time $t$. Relative Relationship Cube is a time series of Relative Relationship Matrices, which is organized in three-dimensional space (two dimensions in space and one dimension in time) according to time. By replacing coordinates with Relationships, RSMO does not require much additional computation to build relationships among objects from the coordinates in traditional STCs and provides a natural approach to analyze a moving object’s behavior.

Fig. 3 provides an example of RSMO’s organization. Fig. 3(a) illustrates the spatial trajectories of five moving objects (Objects labelled as A, B, C, D and E). Fig. 3(b) illustrates a right-handed coordinate system for detecting objects’ Relative Location from each object’s view. Fig. 3(c) shows a geometric transformation method from Relative Locations to
Fig. 3(d) is the derived Relative Relationship Matrix among objects A, B, C, D and E when using the method in Fig. 3(c). In this matrix, the mutual distances and directions (i.e., $\text{Dis}_{A\rightarrow B}$, $\text{Ang}_{A\rightarrow B}$) of moving objects are organized akin to x and y coordinates. In this matrix, rows represent "Reference" objects and columns represent their detected "Target" objects. Fig. 3(e) shows a Relative Relationship Matrix that includes all Relative Relationship Matrices sorted by Time. The proposed model replaces the orthogonal coordinate axes (X, Y) in the STC with two object axes that represent reference and target objects to store their relationships.

Recording the changing relationships between moving objects is insufficient because this process only allows relationship comparison between two different moving objects and cannot support transformation with absolute space to match the computational tasks in current GIS tools. To solve these problems, an additional object R (called the initial reference object), which is a fixed coordinate in absolute space, was added to RSMO (the object R in Fig. 3(b)). This object represents a fixed location acting as a local reference system. By comparing the location change in this local reference system, each object’s motion relative to itself can be derived. After determining the coordinates in absolute space, this object also facilitates transformation between relative space and absolute space. The details of this procedure will be introduced in 3.2.3. This Relative Relationship Cube forms the basic structure of RSMO.

### 3.2. Basic operators of RSMO

This section introduces a set of basic operators for the implemented GIS functions. Those operators are organized into six classes according to the type of relationship outcome.

1. Query processing is an operator to get the direct relative relationship in terms of relative distance and angle between objects and time.
2. Group relative dynamic calculation is an operator to derive the relative dynamic characteristics of the group.

| Operator | Parameters | Function form | Result | Meaning | Example |
|----------|------------|---------------|--------|---------|---------|
| Point extraction | 1. Reference object 2. Target object 3. Time | $E(\text{Ref}, \text{Tar}, t, \ast)$ | Point at a designated position in a cube | Query the relationship between the designated objects at the designated time | $\text{Out} = E(A, B, A_1, \ast)$ |
| Time drilling | 1. Reference object 2. Target object | $E(\text{Ref}, \text{Tar}, \ast, \ast)$ | Line with a designated reference and target object in a cube | Query the changing progress of the relationship to the designated objects | $\text{Out} = E(A, B, A_1, \ast)$ |
| Target drilling | 1. Reference object 2. Time | $E(\text{Ref}, \ast, t, \ast)$ | Line with a designated reference object and time in a cube | Query the designated object’s relationships with other objects | $\text{Out} = E(A, B, A_1, \ast)$ |
| Relationship curvilinear drilling | 1. Reference object 2. Relationship | $E(\text{Ref}, \ast, \ast, R)$ | Line with a designated reference object and relationship in a cube | Query the objects that match the designated relationships with the designated object | $\text{Out} = E(A, B, A_1, \ast)$ |
| Time cutting | 1. Time | $E(\ast, \ast, t, \ast)$ | Face at a designated time in a cube | Query the relationships between all objects at the designated time | $\text{Out} = E(A, B, A_1, \ast)$ |
| Reference cutting | 1. Reference object | $E(\text{Ref}, \ast, \ast, \ast)$ | Face with a designated reference object in a cube | Query the changes in all the objects’ relationships relative to A | $\text{Out} = E(A, B, A_1, \ast)$ |

Note: In query processing, the required parameters of the corresponding Operator are shown in the Parameters column. Function form expresses the operator in a formula. Different operators are described and expressed by $E(\text{Ref}, \text{Tar}, t, R)$ uniformly. The four parameters in the expression are the reference object (Ref), target object (Tar), time value (T) and relationship (R). “Out” is the result of the operator. Result indicates the form of the Operator’s output in the data model. The meaning of each operation is explained in Meaning, and its application is introduced through examples under Example.
Transformation between absolute space and relative space is an operator to transform the data between absolute space (e.g. co-ordinates) and relative space (e.g. relative distance and angle).

Initial reference object transformation is an operator to change the reference object of Relative Relationship Cube, and update all relative relationships of all objects.

Attribute transformation is an operator to derive the relationship attributes based on Relative Relationship Cube, such as, closeness.

Relative relationship dynamic pattern transformation is an operation to derive the relative dynamic pattern between object, such as moving trends of them.

These operators are expanded and developed from STCs to adapt the features of a moving object’s motion in relative space. The following subsections will describe these classes in detail.

3.2.1. Query processing

Query processing searches for the relative mutual relationships of moving objects and their changes. The six operators of query processing are explained as follows:

(1) **Point extraction** retrieves the relationships between designated objects at specific times. This operator is a basic function for querying and analysis.

(2) **Time drilling** retrieves the dynamic processes of the relationships between designated objects.

(3) **Target drilling** extracts the specific reference object’s relationships with other target objects at a designated time.

(4) **Relationship curvilinear drilling** retrieves the target objects that present a specific relationship with the designated reference object. This operator can be used to extract the reference object’s behavior relative to other objects, such as the interactions between two moving objects if “distance < 2m”. The reference object’s interactions can be expressed as a planar 3D curve that consists of target objects for each time.

(5) **Time cutting** retrieves all the objects’ mutual relationships at any designated time.

(6) **Reference cutting** retrieves the movement of all objects relative to the reference object.

Detailed descriptions of these query-processing operators are listed in Table 2:

| Operator | Parameters | Function form | Result | Meaning | Example |
|----------|------------|---------------|--------|---------|---------|
| Time flattening | 1. Time axis | Flattening (Time,Fun()) | Matrix to store the result of the function | Set different flattening functions to obtain the average values of attributes between all the objects. | Query the object pair with the minimum distance between all pairs during the entire period. Expression: Out = Min(Flattening (Time, Fun = \( \sum_{i=1}^{n} \frac{\text{dis}_i}{n} \))) |
| Target flattening | 1. Target axis | Flattening (Tar,Fun()) | Matrix to store all the object’s average distances to other objects at every time stamp | Set different flattening functions to obtain each object’s average value of attributes relative to other objects. | Query the time stamp when the distance between all the objects is minimal. Expression: Out = Min(Flattening (Tar, Fun = \( \sum_{i=1}^{n} \frac{\text{dis}_i}{n} \))) |

Note: In group relative dynamic calculations, Min in Example is used to find the minimum in the dataset. Process details the procedures in the illustration and provides text to help readers understand this operator. Flattening(axis, Fun()) is used to express a uniform form. Two parameters are required: the axis indicates the input of the operator and Fun() is a function to process the values along the axis. Additionally, Fun() can be set by the user for different purposes.
Table 3: Taxonomy of initial reference transformation with schematic illustrations.

| Operator          | Parameters           | Function form | Result                                                                 | Meaning                        | Example                                                                 |
|-------------------|----------------------|---------------|------------------------------------------------------------------------|--------------------------------|------------------------------------------------------------------------|
| Reference shifting| 1. Initial reference | Shift         | Relative relationship cube with object B0 as the initial reference     | Relative relationship         | Describe all the objects' movement at any time from any object         |
|                   | object (Initial)     | (Initial,    | object as the initial reference object                                 | of objects                   | movement in the view B0 from A0 Expression: Out = Shift(A0, B0)          |
|                   | 2. Transformation   | Trans)        |                                                                         |                               |                                                                         |
|                   | target object (Trans)|               |                                                                         |                               |                                                                         |
| Process           |                      |               |                                                                         |                               |                                                                         |

Notes: In process, the initial reference object is transformed from A₀ (marked in red) to B₀ (marked in green).

in Table 1.

3.2.2. Group relative dynamic calculation

Group relative dynamic calculation computes the group relative dynamic characteristics (i.e., average distance, interaction frequency in social media, and the closeness of social relationships) based on the relationship changes of all objects. The two operators Time flattening and Target flattening are explained as follows:

(1) **Time flattening** computes the relative dynamic statistical characteristics between each pair of objects during the entire period and retrieves a matrix that contains the results. For instance, the average distance is calculated through a statistics function, and the unit in the resulting matrix shows the average distance between each pair of objects, which infers the closeness of their relationship during the entire period.

(2) **Target flattening** computes the relative dynamic statistical characteristics of all the target objects relative to each reference object at each time stamp and retrieves a matrix that contains the results. For example, if we want to calculate the interaction frequency, we can obtain a result matrix that indicates the interaction changes of each reference object relative to other objects over time.

Detailed descriptions of these group relative dynamic calculations are listed in Table 2.

3.2.3. Transformation between absolute space and relative space

The purpose of this operator is to build a transformation method between absolute space and relative space. This operator is used to adapt current GIS modules because most GIS software analyzes moving objects only in absolute space.

If we know the coordinates \((x_A^{\text{absolute}}, y_A^{\text{absolute}}), (x_B^{\text{absolute}}, y_B^{\text{absolute}})\) and movement-direction azimuth \(\theta_A^{\text{absolute}}, \theta_B^{\text{absolute}}\) of two objects A and B in absolute space, we can calculate the relative coordinate \([x_B^{\text{relative}}, y_B^{\text{relative}}]\) of B relative to A in relative space with equation (1):

\[
\begin{bmatrix}
x_B^{\text{relative}} \\
y_B^{\text{relative}}
\end{bmatrix} =
\begin{bmatrix}
\cos \theta_A^{\text{absolute}} \sin \theta_B^{\text{absolute}} - x_A^{\text{absolute}} \\
-\sin \theta_A^{\text{absolute}} \\
\end{bmatrix}
\begin{bmatrix}
x_B^{\text{relative}} \\
y_B^{\text{relative}}
\end{bmatrix} =
\begin{bmatrix}
x_A^{\text{absolute}} - x_B^{\text{absolute}} \\
y_A^{\text{absolute}} - y_B^{\text{absolute}}
\end{bmatrix}
\]

Based on \([x_B^{\text{relative}}, y_B^{\text{relative}}]\) we can calculate the relative distance of objects A and B with \(D_{B,A} = \sqrt{(x_B^{\text{relative}})^2 + (y_B^{\text{relative}})^2}\) and infer the relative angle of objects A and B with \(\text{Ang}_{B,A} = \arctan \left( \frac{y_B^{\text{relative}}}{x_B^{\text{relative}}} \right)\), which are explained in Fig. 2(c).

During transformation from relative space to absolute space, we can set an initial reference object \((x_A^{\text{absolute}}, y_A^{\text{absolute}})\) and azimuth \(\theta_A^{\text{relative}}\). If we know the distance (\(D_{B,A}^{\text{relative}}\)) and angle (\(\text{Ang}_{B,A}^{\text{relative}}\)) relative to \(x_A^{\text{absolute}}, y_A^{\text{absolute}}\), we can calculate the coordinates of A \((x_A^{\text{absolute}}, y_A^{\text{absolute}})\) in absolute space by using the following three steps:

(1) The movement-direction azimuth \(\theta_B^{\text{relative}}\) of A is determined by equation (2):

\[
\theta_B^{\text{relative}} = \text{Ang}_{B,A}^{\text{relative}} - \text{Ang}_{B,A}^{\text{absolute}}
\]

(2) The increment in the X and Y directions (\(\Delta x, \Delta y\)) in absolute space can be calculated with equation (3) by distance decomposition:

\[
\Delta x = D_{B,A}^{\text{relative}} \cdot \cos(\theta_B^{\text{relative}}), \quad \Delta y = D_{B,A}^{\text{relative}} \cdot \sin(\theta_B^{\text{relative}})
\]

(3) Based on (\(\Delta x, \Delta y\)), the coordinates of A in absolute space can be expressed with Eq. (4):

\[
x_A^{\text{absolute}} = x_A^{\text{absolute}} + \Delta x, \quad y_A^{\text{absolute}} = y_A^{\text{absolute}} + \Delta y
\]

3.2.4. Initial reference transformation

Initial reference transformation transforms the relationship of any moving object with another object to that of another object. This process changes the analysis view between objects, which could support the relative space-based analysis of any individual object in parallel. A detailed description of initial reference transformation is listed in Table 3. Below, we present an example to explain this process.

An example is introduced to explain the transformation process in Fig. 4(a). A₀ is the initial reference object of object A at t₀. If we want to shift the initial reference object from A₀ to B₀ (object B at t₀), where B₀ is the new initial reference object, we compute all the objects’ unknown
relationships with $B_0$. For example, Fig. 4(a) shows the unknown mutual relationships $B_0-A_1$ and $B_0-C_1$ in the new cube’s matrix.

The mutual relationship between $B_0$ and $C_1$ can be derived by the following steps.

First, the coordinates in absolute space are calculated with the transformation operator between absolute space and relative space from Section 3.2.3.

Second, the distance between $B_0$ and $C_1$ is calculated by the law of cosines. The distance between $B$ and $C$ ($\text{Dis}_{B0, C1}$ and $\text{Dis}_{C0, B0}$) can be calculated with Eq. (5):

$$\text{Dis}_{B0, C1} = \text{Dis}_{C0, B0} = \sqrt{\text{Dis}_{B0, B1}^2 + \text{Dis}_{C0, C1}^2 - 2 \cos \alpha \times \text{Dis}_{B0, B1} \times \text{Dis}_{C0, C1}}$$

Finally, the angle between $B_0$ and $C_1$ is calculated by the following rotation method to ensure uniqueness.

1. The rotations of $B_0$ and $C_1$ relative to object $A_0$ ($\text{yaw}_{A0, B0}$ and $\text{yaw}_{A0, C1}$) are calculated with equation (6):

$$\text{yaw}_{A0, B0} = \text{Ang}_{A0, B0} - \text{Ang}_{A0, A0}$$
$$\text{yaw}_{A0, C1} = \text{Ang}_{A0, C1} - \text{Ang}_{A0, A0}$$ (6)

2. The coordinates of $B_0$ and $C_1$ relative to $A_0$ are computed with equation (7):

$$x_{B0} = \text{Dis}_{B0, A0} \cos (\text{Ang}_{A0, B0})$$
$$y_{B0} = \text{Dis}_{B0, A0} \sin (\text{Ang}_{A0, B0})$$
$$x_{C1} = \text{Dis}_{C1, C0} \cos (\text{Ang}_{A0, C1})$$
$$y_{C1} = \text{Dis}_{C1, C0} \sin (\text{Ang}_{A0, C1})$$ (7)

3. The angle between $B_0$ and $C_1$ in the reference of $A_0$ ($\text{Ang}_{B0, C1}$ and $\text{Ang}_{C1, B0}$) is computed with equation (8):
Table 4
Taxonomy of attribution transformation with schematic illustrations.

| Operator  | Parameters | Function form | Result |
|-----------|------------|---------------|--------|
| Labeling  | Relationship classifier function: Fun() | Label(Fun()) | Label each relative relationship bin with new attribute of closeness based on Fun() |
| Filtering | Filtering function: Fun() | Filter(Fun()) | Obtain the relative relationship bins whose attribute of closeness is intimate |

Note: Label(Fun()) and Filter(Fun()) are used to express uniform forms. The parameters of Fun() in Labeling is a function to acquire new attributes based on the relative relationships of bins. And Fun() in Filtering is used to select the relative relationship bins based on the new attribute by Labeling (e.g., closeness is intimate). Additionally, Fun() in two operators can be set by user for different purposes.

\[
\begin{align*}
\text{Ang}_{\text{dis}}^{\text{Area}} &= \text{atan2}(\text{Dis}_{\text{Ang}_{\text{C}_{1},\text{B}}}, \text{sin}(\text{Ang}_{\text{C}_{1},\text{B}}) \cdot \text{cos}(\text{Ang}_{\text{C}_{1},\text{B}})) \\
\text{Ang}_{\text{dis}}^{\text{Area}} &= \text{atan2}(\text{Dis}_{\text{Ang}_{\text{C}_{1},\text{B}}}, \text{sin}(\text{Ang}_{\text{C}_{1},\text{B}}) \cdot \text{cos}(\text{Ang}_{\text{C}_{1},\text{B}})) \\
\end{align*}
\]

(4) \(\text{Ang}_{\text{dis}}^{\text{Area}} \) and \(\text{Ang}_{\text{dis}}^{\text{Area}} \) can be expressed as

\[
\begin{align*}
\text{Ang}_{\text{dis}}^{\text{Area}} &= \text{atan2}(\text{Dis}_{\text{Ang}_{\text{C}_{1},\text{B}}}, \text{sin}(\text{Ang}_{\text{C}_{1},\text{B}}), \text{cos}(\text{Ang}_{\text{C}_{1},\text{B}})) \\
\text{Ang}_{\text{dis}}^{\text{Area}} &= \text{atan2}(\text{Dis}_{\text{Ang}_{\text{C}_{1},\text{B}}}, \text{sin}(\text{Ang}_{\text{C}_{1},\text{B}}), \text{cos}(\text{Ang}_{\text{C}_{1},\text{B}})) \\
\end{align*}
\]

Note: Label(Fun()) and Filter(Fun()) are used to express uniform forms. The parameters of Fun() in Labeling is a function to acquire new attributes based on the relative relationships of bins. And Fun() in Filtering is used to select the relative relationship bins based on the new attribute by Labeling (e.g., closeness is intimate). Additionally, Fun() in two operators can be set by user for different purposes.

Table 5
Taxonomy of relative relationship dynamic pattern transformation with schematic illustrations.

| Operator | Parameters | Function form | Result |
|----------|------------|---------------|--------|
| Trending | Trend() | | Relative relationship bins with the attributes of motion-trend information: Obtain the relative speed, relative movement direction and other relational motion attributes. |
| Matching | 1. Specific pattern (Pattern, S) 2. Trend sequence (trend) | Match(Pattern, S, trend) | True or False which indicates whether the Trend is matched with Pattern S. Judge whether the relative relationship meet the specific pattern (e.g., moving closer) |

Note: In Trending, there is no parameter for Trend() because motion features are computed by comparing the change of relationships, which are stored in the model. In Matching, the parameter of trend is the result of Trending. The parameter of Pattern S is a relative relationship bins series to represent movement pattern (e.g., moving closer), which is defined by user. In the Result of Matching, the result of Match(Pattern, S, trend) is False because the relationship of trend A, B in T1 is moving away, which is not matched with the Pattern S (in black cell).
based on the conditions or classifier set from Fun().
(2) Filtering is used to remove some bins from the results of the Labeling step.
(3) Detailed descriptions of these attribution transformations are listed in Table 4.

3.2.6. Relative relationship dynamic pattern transformation
The goal of relative relationship dynamic pattern transformation is to analyze the motion features of moving objects in relative space. Two common operators are explained as follows:

(1) Trending computes motion features (such as distance changes, angle changes, relative speeds and relative acceleration) by comparing relationship changes between each pair of objects.
(2) Matching discovers relative motion patterns with the condition of a movement pattern, such as accompaniments and trackers.

The detailed descriptions of these relative relationship dynamic pattern transformations are listed in Table 5.

4. Implementation

4.1. Implementation of the proposed model

Fig. 5 shows a UML class diagram of the proposed RMSO model. In this diagram, the class Relative Relationship Cube maintains a list of Relative Relationship Matrices by the time value, and Relative Relationship Matrix manages a series of Relative Relationship Bins, each containing a relationship (distance and angle) between two Objects at certain times. These classes are used to manage the original relative data.

This diagram implements the operators from Section 4 as functions of a Relative Relationship Cube. The results of these operators (Point extraction, Time drilling, Target drilling, Relationship curvilinear drilling, Time cutting, Reference cutting, Time flattening, Target flattening, Labeling, Filtering and Reference shifting) can be expressed as Relative Space Points, Relative Space Lines, Relative Space Surfaces, and Relative Space Cubes. In particular, a Relative Trend Bin is defined as a subclass of Relative Space Points. Relative Trend Bins offer attribute functions to describe their motion trends according to changes in relationships. Relative Trend Lines contain a list of Relative Trend Bins and are used to achieve a series of Relative Trend Bins with specific patterns.

A prototype (Fig. 6) was developed to implement the proposed RMSO model. All the functions store, manage and analyze relative space data were encoded into a dynamic link library within C++ environment. The visualization of this prototype was developed by the QGIS (Quantum GIS) framework in a Python environment. Two typical analysis modules were implemented, namely, query processing and relative relationship dynamic pattern transformation. The query processing contained all six operators (Point extraction, Time drilling, Target drilling, Time curvilinear drilling, Time cutting and Target cutting). Two relative relationship dynamic pattern matching algorithms (accompaniments and trackers) were implemented in this prototype.

4.2. Query-processing algorithm

A query algorithm called Query processing is described here to for retrieving relative relationship in relative space. The main idea of this algorithm are to organize the relative relationship as three dimensions, namely, Reference object, Target object and Time, which is based on model’s structure, and then to index relative relationship by corresponding objects and time. By using a composite index consisting of Reference object, Target object and Time, this algorithm can quickly find out the relative relationship completely covered by the given restricted objects and time. In the query, this algorithm retrieves relationships by accessing the relative relationship according to the corresponding composite index of given object and time, instead of extracting each relative relationship bins of reference or target object one by one in the cube. Therefore, this algorithm can be integrated with all the operators shown in Section 3.2.1 and constrained by six parameters, namely, Cube, Query type, C ref, C tar, C T, and rel, which are explained in Algorithm 1. Cube is a structure of map, whose key consists of a reference and target object, while value is also a map whose key is the time and value is the relationship. The detail of the Query processing algorithm is described in Algorithm 1.

Algorithm 1: Query-processing (Cube, Query type, C ref, C tar, C T, rel)

| Line | Description |
|------|-------------|
| 1    | let Q be a sequence of bins of cube. Q contains pairs of the form < objects_pair, relptr >, where objects_pair is the index on attribute ref and tar in a pair form and relptr is a map structure whose time is the key and relationship is the value. |
| 2    | let Count be the number of objects in the scenario. |
| 3    | let Cand is the output in a list form |
| 4    | while Q is not empty do |
| 5    | s= < objects_pair, relptr > = Q.dequeue; |
| 6    | switch (Query.type) |
| 7    | case Point extraction: |
| 8    | if objects_pair -> first = C ref then break; |
| 9    | else if objects_pair -> second = C T then break; |
| 10   | else if relptr -> key = C T then break; |
| 11   | else s= < objects_pair, relptr > ; |
| 12   | break; |
| 13   | case Time drilling: |
| 14   | if objects_pair -> first = C ref then break; |
| 15   | else if objects_pair -> second = C tar then break; |
| 16   | else s= < objects_pair, relptr > ; |
| 17   | break; |
| 18   | case Target drilling: |
| 19   | if objects_pair -> first = C ref then break; |
| 20   | else if relptr -> key = C T then break; |
| 21   | else s= < objects_pair, relptr > ; |
| 22   | break; |
| 23   | case Time curvilinear drilling: |
| 24   | if objects_pair -> first = C ref then break; |
| 25   | else if relptr -> value = rel then break; |
| 26   | else s= < objects_pair, relptr > ; |
| 27   | break; |
| 28   | case Time cutting: |
| 29   | if relptr -> key = C T then break; |
| 30   | else s= < objects_pair, relptr > ; |
| 31   | break; |
| 32   | case Reference cutting: |
| 33   | if objects_pair -> first = C ref then break; |
| 34   | else s= < objects_pair, relptr > ; |
| 35   | break; |
| 36   | Cand.append(s); |
| 37   | return Cand; |

When comparing this algorithm with traditional GIS querying in absolute space, this algorithm can avoid many redundant computational address coordinates, which can improve the efficiency of the relative relationship querying. We will test this approach in Section 5.5.

4.3. Relative relationship dynamic pattern matching algorithm

Algorithm 2, which is called Pattern matching, illustrates the process of searching a specific relative pattern in relative space. Dynamic pattern is represented as a space-time trajectory (Bao et al., 2017), which is consist of specific locations in absolute space (shown in Fig. 7). Given a specific pattern (Pattern S), each object’s distance to the Pattern S needs to be calculated to measure its similarity to Pattern S. Finally, the
trajectory, whose distances are all in matching threshold (in Fig. 7), would be got as the outcome of Pattern matching in absolute space. In relative space, dynamic pattern is represented as a relative relationship bin series, which is composed of bin with specific relative distance. In this algorithm, the bins with each object’s relative distance to Pattern_S can be directly got by reference cutting at first. Then, they are labeled according to whether their relative distance is in matching threshold. Finally, the relative relationship bin series are obtained, which match the Pattern_S in all time. This algorithm provides a more efficient way for relative relationship dynamic pattern matching. It simplifies the spatial computation to measure the similarity on basis of reference cutting and labeling in the proposed model.
The details of the Pattern matching algorithm are described in Algorithm 2.

Algorithm 2: Pattern matching (Cube, C_ref, Pattern_S)

Input: Cube – the relationships between all objects in a relative space–time cube form; C_ref – the reference object, an index on ref of a relative relationship bin in Cube; Pattern_S – a specific relative relationship pattern to be found, which is expressed as a time series that consists of a triple of time, rel.

Output: the object in cube, which produces a trend of Pattern_S relative to C_ref.

1. let Q be a sequence of bins in the cube. Q contains pairs of the form < objects_pair, relptr >, where objects_pair is the index on attribute ref and tar in a pair form and relptr is a map structure whose time is the key and relationship is the value;
2. let ObjectsList be the set of objects that contains the codes of all the objects in the scenario;
3. let TimeStamps be the total number of time stamps in the data;
4. let * be any value;
5. let Labeling be the function that get the relative relationship bins, which match the Pattern_S;
6. Cand is output;
7. for each object O in ObjectsList do:
8.  Trj = Query processing(Cube, Time Drilling, C_ref, O, *, *);
9.  for each value T in TimeStamps do:
10.  Trends_S = Pattern_S[T];
11.  bin_O = Trj[T];
12.  if Label (Trends_S, bin_O) == null then break;
13.  else continue;
14.  Cand.add(O);
15. return Cand;

This algorithm allows users to search objects with specific motion patterns relative to dynamic reference objects. This advantage can help users discover several meaningful behaviors of objects that are hidden in a moving group, for example, being followed.

5. Case studies

The proposed RMSO model was tested with three application scenarios, such as epidemic spreading, tracker finding, and the motion-trend derivation of nearby crowds. Each case study was used to demonstrate the feasibility and advantages of the proposed model.
5.1. Dataset

This paper collected pedestrians’ (experimental subjects) walking trajectories on the street in the three experimental scenarios. A Lenovo Phab 2 Pro phablet was used to record their trajectories in both absolute and relative space. This device is openly available and equipped with sensors for motion tracking. This phablet can record the position and rotation angle relative to the starting pose of the device. The GPS receiver recorded the motion in absolute space and relative space. Fig. 8 shows all 34 trajectories of the experimental subjects walking along a road in Wuhan City, China. The detailed information includes the following fields in Table 6:

- "UserID" is the unique identifier for each object.
- "Timestamp" is the time when the location was recorded.
- "Longitude" and "Latitude" were used to record the pedestrians’ location in absolute space with GPS receiver.
- "X increment" and "Y increment" show the relative locations in the reference frame that was built by the user’s starting pose.
- "Rotation angle" records the change in angle relative to the initial pose in a counterclockwise direction.
- "Initial azimuth" is the angle of the user’s face relative to the north.

The collected data needed to be pre-processed before saved into the proposed model as the following steps.

(1) Derive relative distance and angle between all objects. First, the relative distance could be computed by the operator of transformation between absolute space and relative space with the coordinates in the fields of Longitude and Latitude in collected data. Then, the relative angle was computed based on the movement-direction azimuth ($\phi_{relative}$), which was derived from Rotation angle ($\phi$) and Initial azimuth ($\alpha$) according to the equation (13).

$$\phi_{relative} = \phi + \alpha$$

(13)

(2) Set the initial reference object for the model. Initial reference object was set as object 2 at t0 in this experiment by computing all objects’ relationships to object 2 at t0. The operator of transformation between absolute space and relative space was implemented for this task under the condition that each objects’ coordinates (Longitude and Latitude), Rotation angle and Initial azimuth are known in collected data.

5.2. Epidemic spread and close-contact querying

The first scenario involved finding the person who contacted with an epidemic carrier. In the research community of epidemiology, the spread of diseases is closely related to the spatial and social activities of patients, and the propagation of diseases greatly depends on the trajectory of a patient's social activities. Thus, tracing the back-propagation path of a virus and finding close contacts are meaningful to prevent the spread of diseases. This experiment assumed a severe acute respiratory syndrome (SARS) carrier’s walking path to find its close contacts. SARS is a viral respiratory disease of zoonotic origin that is caused by the SARS coronavirus (SARS-CoV). The primary route of transmission for SARS is the contact of mucous membranes with respiratory droplets or fomites (World Health Organization, 2003). The research about respiratory droplet transmission shows that the largest droplets will totally evaporate before falling 2m away (Xie et al., 2007). Therefore, a person will be identified as a close contact if the distance to the carrier (whose ID is 2 in Fig. 9) is less than 2m. Therefore, this paper uses 2m as the parameter for querying close contacts.

This query was implemented by combining the Reference cutting and Labeling operators (shown in Fig. 10(a) and (b)). First, the relative relationships (distance) of all the experimental subjects relative to the carrier were derived by Reference cutting. Here, a matrix (shown in Fig. 9) was used to show the change in distance to the carrier for each experimental subject; blue means a closer distance to the carrier, and red means a farther distance to the carrier. Then, the relative relationship bin indicating a close contact (in green) was determined by Labeling for distances below 2m. Thus, the subjects (19 and 32) were found as close contacts.

Reference cutting (a) and Labeling (b) are used to search the relative relationship bins containing close contacts in the data model, which are labeled in red. The close contacts’ relative locations to the carrier are marked in (c). And their location in absolute space are shown in (d) by the operator of transformation between relative space and absolute space. By using the transformation between absolute space and relative space.
In the research community of crime, tracking is a basic behavior for further crime (Beauregard et al., 2010). Finding trackers are helpful to take caution actions to prevent crime. Usually, the tracker’s behavior is similar to other normal pedestrians in the crowd. Therefore, victims and public security organizations may experience difficulty discovering a tracker in absolute space.

In this experiment, we assumed that the subject 2 carried large amounts of cash, and he had to be aware of potential trackers. To remain unnoticed, trackers usually stay within a certain range from the target to neither be found nor lose the target. This distance range is a critical parameter to build specific patterns for trackers. According to the sociological theory (Moore et al., 2010), trackers are easily found if

---

Fig. 9. Distances of all subjects relative to the carrier. The subjects that are labeled in green bins indicate close contacts. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 10(c) and (d) show the two locations of the subjects 32 and 19 in relative space and absolute space. One was much close to the carrier, the first being at 1.94 m at 16:17:20 and the other at 1.44 m at 16:18:30. According to these querying results, public health agencies should take actions to control these two people. Thus, this query is very helpful for public health agencies to control epidemic spreading if embedded into any real-time GIS analytical systems.

5.3. Tracker finding

The second scenario was to find any tracker of a subject in a crowd.
their distance to the victim is less than 3 m. Therefore, this study set the parameter of 3 m as the lower distance limit for this query. On the other hand, the upper distance limit must guarantee that the target is within the tracker’s sight. This study set the upper distance limit as 42 m to ensure that the tracker did not lose the target. This parameter was calculated according to the waiting time of intersections in this study area (30 s). Therefore, \(30 \text{s} \times 1.4 \text{m/s} = 42 \text{m}\), where 1.4 m/s is the normal average walking speed (Fitzpatrick et al., 2006). Sometimes, the distance is probably larger than 42 m, which may cause the tracker to lose the target; thus, the tracker will speed up to follow the target. In this case, the tracker must find the target within 30 s because of the constraints of traffic lights. The conditions of tracker finding can be expressed as follows:

1. The distance between the tracker and target is always [3 m, 42 m);
2. The time that the distance between the tracker and target is not [3 m, 42 m) is below 30 s, which occurs only once while speeding up because of the traffic-light locations in the study area.

Three main operators (Reference cutting, Labeling, Matching) were used to find the tracker by meeting the above conditions. Similar to the close-contact querying of epidemic spreading, this study used the Reference cutting operator to find a matrix with all subjects’ relationships relative to 2. Then, this process used the Labeling operator to label the relative distance to all relative relationship bins. Fig. 11 shows the labeled results. In this figure, bins that were labeled in red and yellow had a relative distance less than 42 m, while bins that were labelled in blue had a relative distance larger than 42 m. Fig. 11 illustrates the change in the relative distances between all the subjects and the target. Two related patterns could be derived through Matching: (1) the subject was close to the target most of the time (i.e., 4, 8 and 21), or (2) the tracker lost the target after a very short time (i.e., 6). Fig. 12 shows the speed pattern of this subject. Obvious acceleration was observed between T1 and T3, meeting the second condition of identifying the tracker. Based on the above results, the four subjects were viewed as trackers.

In order to demonstrate the advantages of the proposed approach, this study also shows the trajectories of 4 and 6. By the operator of transformation between absolute space and relative space, the trajectories of these individuals and the target did not present obvious abnormal features compared to other trajectories in absolute space (Fig. 13(a)). However, their relative trajectories relative to the target presented an obvious accompaniment feature in relative space (Fig. 13(b) and (c)). Therefore, this proposed approach can mine hidden patterns that are ignored by traditional GIS data models.

5.4. Deriving the motion trends of nearby crowds

The third scenario was to derive the motion trends of nearby crowds. The motion trend of a nearby crowd is an important feature of surrounding dynamics and has a large influence on decision-making processes (Fang et al., 2003). This concept is necessary for individuals to understand nearby motion trends to avoid heavy congestion or stampedes. This experiment assumed the subject 2 as a walker who

---

**Fig. 11.** Distances of all the subjects relative to 2.

**Fig. 12.** Acceleration of 6, which was positive when the target was lost.
hoped to be aware his surroundings, for example, how many people moved closer to or farther from him and whether they were accelerating towards him.

Five main operators (Reference cutting, Trending, Labeling, Filtering and Time flattening) were used to implement this application. First, the relative relationships of all the experimental subjects relative to the

![Diagram of trajectories](image-url)
walker were derived by using *Reference cutting*. Then, a matrix that contained the motion trend (relative distance change and relative acceleration) was calculated through the *Trending* operator. This matrix was labeled with a legend of colors in Fig. 14 through the *Labeling* operator, which illustrates changes in the relative trend motion of the other subjects compared to the walker. In Fig. 14, the red bins in the matrix mean that these subjects are accelerating towards the walker, orange bins mean that the subjects are decelerating towards the walker, gray bins mean that the subjects are accelerating away from the walker, blue bins mean that the subjects are decelerating away from the walker and yellow bins mean no change in distance and speed. The gray, blue and yellow relative relationship bins were removed with the *Filtering* operator. Fig. 15(a) shows the results after this operation. Then, the number of bins where the subjects moved towards the walker was counted by *Time flattening*, which counted any red or orange bins. The updated result is shown in Fig. 15(b), indicating that more than half of the individuals were moving closer to the walker from T1 to T27 (16:14:05–16:16:15). Similarly, Fig. 15(c) shows the number of subjects that were accelerating towards the walker at each time, indicating that more than half of the subjects were accelerating towards the walker at eight times (T2, T4, T9–T11, T12, T15, T17, T20–T23 and T26–T27).

This study transformed these bins from relative space to absolute space by the operator of transformation between absolute space and relative space, and the locations of surrounding people are shown in Fig. 16(a) and (b) at 16:14:05 and 16:14:40, respectively. By combining all the locations of surrounding people close to the walker, we could find the concentrated area of surrounding people, which is labeled in red in Fig. 16(c). In this case, the walker may have felt uncomfortable and potentially at risk in the crowd road. Setting walker as initial reference object, this model derived the crowd’s motion trends for all subjects by the operator of initial reference transformation. Fig. 17 showed the results of initial reference transformation from object 2 at t0 to object 14 at t16. Those who were moving closer to subject 2 in Fig. 17(a) didn’t show clear motion trend for the new walker (subject 14 at t14) in Fig. 17(b). This method is helpful for people-oriented routing services, such as pedestrian navigation or tourism.

5.5. Performance testing

This study tested the performance of the proposed model in two aspects. One was the time complexity of querying functions; the other was the time complexity between the proposed model and the absolute method in ArcGIS Geodatabase. This study used a dataset (Gramaglia et al., 2014) of vehicle trajectories on two highways around Madrid, Spain. The detailed information in these trajectories contained five fields (time, ID, X position (m), Y position (m) and speed (m/s)). Before the comparison, this study chose 14 datasets for this open-access dataset that contained m subjects for n times, where m = 10, 50, 100, 500, 1000, and n = 50, 100 or 200. Table 7 lists these datasets and the sizes of their storage spaces. All the computational tasks were single-threaded implemented on a DELL desktop (four Intel(R) Xeon(R) processors with CPU E3-1220 v5 @ 3.00 GHz, 16G of RAM and a 64-bit operating system).

Six main functions (*Point extraction, Time drilling, Reference cutting, Querying close contacts, Finding trackers* and *Deriving the motion trends of nearby crowds (Deriving trend]*) were selected to test the time complexity of the proposed approach and the absolute method. The time complexity is shown in Table 7. Overall, the proposed method had a better time complexity than the absolute method.
complexity. Table 7 lists the execution times for the 14 datasets and Fig. 18 illustrates the computation time of each function among these datasets. Both tables show that the computation time of all the selected functions was less than 3 s if the data volume of the datasets was smaller than 1 GB. The execution times significantly increased when the dataset contained more than $5.625 \times 10^7$ records. Among the six functions, Point Extraction and Finding Track spent significantly less time than the other four functions when the records increased to $7 \times 10^7$ in the dataset. This result shows that the proposed model could work effectively with more than 3 GB of data and meet behavior-analysis requirements in scenarios with 1000 subjects.

The second comparison examined the time efficiency between the proposed model and the absolute model (Geodatabase in ArcGIS). The proposed model presents an obvious advantage in terms of conducting three functions (Point extraction, Time drilling, Reference cutting) because these functions only require a simple query, while the absolute model requires additional relative relationship transformation. The other three functions (Querying close contacts, Finding tracker and Deriving the motion trend of crowd nearby) are meaningful to real situations and require a relatively complicated implementation. Therefore, this study compared the second set of three functions that were implemented by the proposed model and absolute model. In ArcGIS, “Buffer” and “Overlay analysis” were used to implement Querying close contacts. The point features of the reference subject’s trajectory were the input features of Buffer analysis in ArcToolBox. The outputs of this operation were face features that represented the close-contact area. Then, the Intersect tool in Overlay analysis could be used to check the status of the subjects’ trajectories inside or outside the close contact area. However, no appropriate tools exist in ArcGIS to directly implement Finding tracker or Deriving the motion trend of crowd nearby. In terms of Finding tracker, we computed each subject’s distance relative to the reference subject at each time via coordinates in Geodatabase and filtered the subjects according to the conditions in Section 5.3. In terms of Deriving the motion trend of crowd nearby, we calculated all the subjects’ relative speeds and relative accelerations relative to the reference subject and statistically counted five types of motion trends, as described in Section 5.4.

Table 8 lists the execution times of the three selected functions for the proposed model and absolute GIS method with 12 of the 14 datasets; these last two datasets were omitted because the absolute GIS method would stop responding if the volume of dataset increased to 3.5 GB. Fig. 19 illustrates the computation time of each function among these datasets. The execution times of the proposed model were approximately 5–50% those of the absolute GIS method for the same function. The computation time of Querying close contacts for the proposed model was approximately 5% that for the absolute GIS method, the most evident improvement from the proposed model. This improvement considerably increased with increasing dataset volume (Fig. 19(a)), and the execution-time difference was $331 \text{ s}$ when the volume was $3.46 \text{ GB}$. The second function was Deriving the motion trend of the crowd nearby, whose time-efficiency improvement was similar to that of Querying close contacts. The execution time of the proposed model was approximately 10% that of the absolute GIS method. Additionally, the improvement from the proposed model became significant if the dataset’s records were larger than $3 \times 10^7$ (Fig. 19(c)). The execution-time difference was $171 \text{ s}$ when the volume was $3.46 \text{ GB}$. The execution time of the third function (Finding tracker) with the proposed model was only 10%–50% that of the absolute GIS method; this improvement was not as evident as that for the previous two functions. In Fig. 19(b), the execution time from the proposed model and absolute GIS method steadily increased as the number of dataset records increased. However, the execution-time difference only reached $8.146 \text{ s}$ when the dataset’s volume was $3.46 \text{ GB}$, an insignificant increase compared to the previous two functions. This result indicates that the proposed model exhibits better efficiency when implementing these
functions compared to the absolute GIS method.

6. Discussion

In this study, the execution times of the three selected functions (Querying close contacts, Finding tracker and Deriving the motion trend of crowd nearby) for the proposed model and absolute GIS method with 12 of the 14 datasets are used to indicate the efficiency improvement of the proposed model. Thus, this section discusses the improvement in respect of model’s structure and operator for the previous three case studies.

(1) In the case of epidemic spread and close-contact querying, this paper analyzed the processes of the proposed model and absolute GIS method for querying close contacts of the carrier. In absolute GIS, this task could be achieved by buffer analysis in ArcGIS software. Corresponding buffer areas related to a carrier had to be built while it was moving all the time in absolute space. However, in the proposed model, there is no need to build many buffer areas to recognize relationships in all time. It only executed once reference cutting to get all object’s relationships to the carrier, no matter how many objects in the scenario. This is fundamental reason for the better performance of our model in this case study. Table 8 and Fig. 19(a) show that more relationships had to be recognized in absolute GIS with the increasing number of objects in the scenario, which leads to the execution time increase more quickly compared with the proposed model.

(2) In the case of tracker finding, the pattern matching approach in absolute space needs to compute the relative relationships of moving objects first, then uses data structure to store the time series of relative relationships between target objects with others, and finally filters the time series of relative relationships with distance constraints to match the pattern of the tracker. It is a complicated computation process in absolute GIS method. The proposed model directly extracted each object’s relationship referred to the target in relative space by reference cutting. Then, the operators of trending and matching were used to find tracker. These computation processes are simple matrix-based computation. Therefore, it could have a better computation performance than the pattern matching approach in absolute space. Results in Table 8 and Fig. 19 (b) demonstrate the execution times by the proposed model is much shorter than that in absolute GIS method.

(3) In the case of deriving the motion trends of nearby crowds, motion trends of crowd nearby in absolute GIS method need to computing each object’s distance to the walker from coordinates in each

| Function | Dataset | Point Extraction | Time Drilling | Reference Cutting | Querying Close Contact | Finding Tracker | Deriving trend |
|----------|---------|------------------|---------------|-------------------|-----------------------|----------------|---------------|
| 1        | m 10    | n 50             | 400 KB        | 0.001             | 0.001                 | 0.001          | 0.001         |
| 2        | 10 100  | 100              | 800 KB        | 0.002             | 0.002                 | 0.002          | 0.002         |
| 3        | 10 200  | 1.56 MB          | 0.003         | 0.002             | 0.005                 | 0.003          | 0.003         |
| 4        | 50 50   | 8.58 MB          | 0.024         | 0.024             | 0.024                 | 0.023          | 0.024         |
| 5        | 50 100  | 17.2 MB          | 0.052         | 0.047             | 0.048                 | 0.047          | 0.047         |
| 6        | 50 200  | 34.7 MB          | 0.1           | 0.097             | 0.099                 | 0.1            | 0.099         |
| 7        | 100 50  | 34.1 MB          | 0.099         | 0.097             | 0.096                 | 0.096          | 0.096         |
| 8        | 100 100 | 68.7 MB          | 0.194         | 0.195             | 0.198                 | 0.193          | 0.195         |
| 9        | 100 200 | 138 MB           | 0.392         | 0.393             | 0.396                 | 0.394          | 0.396         |
| 10       | 500 50  | 874 MB           | 2.556         | 2.61              | 2.611                 | 2.601          | 2.63          |
| 11       | 500 100 | 1.71 GB          | 5.434         | 5.744             | 7.713                 | 5.7            | 5.849         |
| 12       | 500 200 | 3.46 GB          | 14.12         | 15.774            | 15.857                | 16.36          | 16.743        |
| 13       | 1000 50 | 5.42 GB          | 13.583        | 16.256            | 16.063                | 16.406         | 16.542        |
| 14       | 1000 100| 6.87 GB          | 135.835       | 161.159           | 161.995               | 165.192        | 134.467       |

Fig. 18. Curves of the 6 main functions’ execution times with the 14 datasets.
time first, then compares the time series of relative distance to find the moving trend of close or far away, and finally to plot it as a distribution map in absolute space, which supports the choosing of walk routes. This computation task is also an implicated computation process. The proposed model could analyze their distance change characteristic to the walker by the operators of trending and matching, which are also matrix-based computations suitable to computer environment. Table 8 and Fig. 19 (c) shows that the proposed model saved approximately 90% of the execution time compared to absolute space-based GIS method.

In short, the proposed model outperforms the absolute space-based approach in relative motion analysis by simplifying the process of relationships computation and querying.

7. Conclusions

By extending STC in ArcGIS, this paper presents a novel model for storing, managing and analyzing moving objects based on their relationships in relative space. In this model, the X and Y in STC are substituted as the reference and target objects, and the STB in STC are extended to store the data structure related to relative relationship among moving objects. To support the analysis of moving objects, this paper introduces six classes of operators according to the type of relationship outcome, namely, query processing, group relative dynamic calculation, transformation between absolute space and relative space, initial reference object transformation, attribute transformation, relative relationship dynamic pattern transformation. Then, two common used algorithms, relative relationship querying and relative relationship dynamic pattern matching, are introduced on the base of these defined operators. The former algorithm organizes the relative relationship as three dimensions (Reference object, Target object and Time) based on the model’s structure, and queries relative relationship by corresponding objects and time. The later algorithm implements reference cutting and labeling to match each object’s relative motion with specific pattern, which simplifies similarity measure in absolute space. This study collected the trajectories of walking pedestrian and used these data to demonstrate the feasibility of the proposed model. Finally, the proposed model was successfully tested in three real-life scenarios to analyze dynamic and complex relationships between moving objects. The results validated the capabilities of these designed functions and indicated that the proposed model saved up to 50%-95% execution time compared with traditional absolute GIS methods. And the proposed model could be widely applied in the domains of public health, security, individual-based services. Therefore, the contributions of this paper could be summarized as follows:

(1) A relative space-based GIS data model of moving objects (RSMO) is proposed to construct, operate and analyze moving objects’ relative relationships in relative space.

(2) Relative space-based relationship-querying algorithm and relative

Table 8
Comparison of the execution times (unit: second) between the proposed model and absolute GIS method.

| Function                          | Querying close contacts | Finding tracker | Deriving the motion trends of nearby crowds | Absolute | Relative |
|-----------------------------------|-------------------------|-----------------|--------------------------------------------|----------|----------|
|                                  | Size | m | n | 400 KB | 0.02 | 0.001 | 0.015 | 0.001 | 0.016 | 0.001 |
|                                  | 2    | 10 | 50 | 800 KB | 0.06 | 0.002 | 0.02 | 0.002 | 0.02 | 0.002 |
|                                  | 3    | 10 | 200 | 1.56 MB | 0.088 | 0.003 | 0.088 | 0.003 | 0.077 | 0.003 |
|                                  | 4    | 50 | 50 | 8.58 MB | 0.281 | 0.023 | 0.203 | 0.024 | 0.422 | 0.024 |
|                                  | 5    | 50 | 100 | 17.2 MB | 0.422 | 0.047 | 0.422 | 0.047 | 0.768 | 0.047 |
|                                  | 6    | 50 | 200 | 34.7 MB | 1.938 | 0.1 | 0.49 | 0.099 | 1.668 | 0.097 |
|                                  | 7    | 100 | 50 | 34.1 MB | 2.327 | 0.096 | 0.532 | 0.096 | 2.282 | 0.094 |
|                                  | 8    | 100 | 100 | 68.7 MB | 3.264 | 0.193 | 0.625 | 0.195 | 2.372 | 0.191 |
|                                  | 9    | 100 | 200 | 138 MB | 5.112 | 0.394 | 1.994 | 0.396 | 3.656 | 0.384 |
|                                  | 10   | 500 | 50 | 874 MB | 36.07 | 2.601 | 5.657 | 2.63 | 46.47 | 2.627 |
|                                  | 11   | 500 | 100 | 1.71 GB | 108.12 | 5.7 | 11.41 | 5.849 | 92.59 | 5.711 |
|                                  | 12   | 500 | 200 | 3.46 GB | 348.313 | 16.36 | 24.889 | 16.743 | 188.468 | 16.98 |

Fig. 19. Curves of the execution times for “Querying close contacts” (a), “Finding tracker” (b) and “Deriving the motion trends of nearby crowds” (c) with the proposed model (blue line) and Geodatabase (red line). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
dynamic pattern-matching algorithm are proposed on the base of the proposed model.

(3) Three real-life scenarios are implemented to demonstrate the feasibility and usefulness of this proposed model in discovering interesting behavior hidden in crowd (e.g. querying close contact, finding taxi drivers and helping the crowd’s motion trends). Moreover, the computational time of these real-life scenario analysis could save up to 50–95% while compared with absolute space based methods.

The proposed model still has a big disadvantage. The data volume of the proposed data model was much larger than that of the traditional absolute space-based data organization. The compression of these relative data is a challenge. Future research need study efficient compressing algorithms to simplify the data pre-process and reduce the volume of relative data. The handling of multiple types of relationships could also be examined in this model, e.g., social and mental relationships. Powerful operators (i.e., closeness analysis in groups) in the social domain could be integrated into the model to support the analysis of social phenomena. Moreover, high-performance computing technologies such as cloud computing could be incorporated with the proposed model to design more efficient algorithms for relative relationship querying and pattern mining.

Acknowledgement

The research was supported in part by the National Natural Science Foundation of China (Grants 41771473, 41231171), and the Fundamental Research Funds for the Central University.

References

AllMuhiM, F., Durand, N., Quafafou, M., 2018. Detecting behavior types of moving object trajectories. Int. J. Data Sci. Anal. 5 (2–3), 169–187.
Anuo, S., Suzuki, E., 2011. Role-behavior analysis from trajectory data by cross-domain learning. In: Data Mining (ICDM), 2011 IEEE 11th International Conference on, IEEE, pp. 25–30.
Bach, B., Dragicevic, P., Archambault, D., Hurter, C., Carpendale, S., 2017. A descriptive framework for temporal data visualizations based on generalized space-time cubes: generalized space-time cube. Comput. Graph. Forum 36, 36–61.
Bae, J.-G., Chung, C.-D., Zhou, Y., 2017. Algorithms for Mining Human Spatial-Temporal Behavior Pattern from Mobile Phone Trajectories. In: 2017 2nd International Conference on Computer Science and Technology, pp. 765–773.
Bastille-Rousseau, G., Gibbs, J.P., Frair, J.L., Cabrera, F., Rousseau, L.-P., Wilkecki, M., Kümmerl, F., Blasø, S., 2017. Animal movement in the absence of predation: environmental drivers of movements strategies in a partial migration system. Oikos 126 (7), 1004–1019.
Bearegard, E., Rebocho, M.F., Rossmo, D.K., 2010. Target selection patterns in rape. J. Invest. Psychol. Offender Profil. 7 (2), 137–152.
Besse, P.C., Guillouet, B., Loubes, J.M., Royer, F., 2018. Destination prediction by trajectory distribution-based model. IEEE Trans. Intell. Transp. Syst. 19 (8), 2470–2481.
Bogorny, V., Renso, C., Aquino, A.R.D., Siqueira, F.D.L., Alvares, L.O., 2014. Constant–a posed model to design more efficient algorithms for relative relation-
