## **Research Article**

Miao Wang\*, Zhenxi Fang, Weiguang Liu, Xiaotong Wang, Xiaodong Liu, and Songyang Li

# Computing the inverse of cardinal direction relations between regions

https://doi.org/10.1515/jisys-2022-0202 received May 20, 2022; accepted October 23, 2022

**Abstract:** The inversion of directions is an important operation with directions which plays an important role in qualitative spatial reasoning and spatial queries. In this work, we address on the inversion operation of the basic cardinal direction relations in the model of Goyal. The direction relation matrix model proposed by Goyal is a projection-based model for spatial direction relations between regions. This model is simple in calculation and easy to carry out formal reasoning, which is considered as currently one of the most excellent models for representation and qualitative reasoning with cardinal direction relations in two-dimensional space. This work aims to realize the automatic inference and calculation of the inverse of the basic cardinal direction relations in the model of Goyal and further to improve the ability of spatial reasoning and spatial analysis of spatial database. In order to avoid the complicated manual reasoning, an algorithm for automatically performing the inverse operation on this model is devised by means of the operations of direction relation matrix. Theorems are provided to prove formally that our algorithm is correct and complete, which is also verified by comparing the result of our algorithm with that of manual reasoning for each basic cardinal direction relation. This study realized the automatic inference and calculation of the inverse of the basic cardinal direction relations in the model of Goyal and further improved the ability of spatial reasoning and spatial analysis of spatial direction relation.

Keywords: inversion, cardinal direction relations, direction relation matrix, spatial database

# **1** Introduction

Spatial direction relation as one of the basic concepts in spatial cognitive, is used to describe the position information with one object relative to another which reflects the ordering relations between spatial objects [1]. It is an indispensable part of spatial relation theory as well as topological relation and qualitative distance relation. With the further application of direction relations, people are no longer satisfied with simply describing and storing, but also need the ability of spatial reasoning to deal with the inference of the unknown information and the consistence checking of the existing information for direction relations. In recent years, qualitative spatial reasoning with direction relations has received a lot of attention in the areas

<sup>\*</sup> **Corresponding author: Miao Wang,** Department of Computer Science, Henan University of Engineering, No. 1, Xianghe Road, Longhu, Zhengzhou, Henan, P.R. China; Department of Computer Science, Zhongyuan University of Technology, No. 41, Zhongyuan Road, Zhongyuan District, Henan, P.R. China, e-mail: wmscan@tom.com

**Zhenxi Fang:** Department of Computer Science, Zhongyuan University of Technology, No. 41, Zhongyuan Road, Zhongyuan District, Henan, P.R. China, e-mail: fieldlay@163.com

**Weiguang Liu:** Department of Computer Science, Zhongyuan University of Technology, No. 41, Zhongyuan Road, Zhongyuan District, Henan, P.R. China

Xiaotong Wang: Department of Computer Science, Central South University, 932 South Lushan Road, Changsha Hunan, P.R. China Xiaodong Liu, Songyang Li: Department of Computer Science, Henan University of Engineering, No. 1, Xianghe Road, Longhu, Zhengzhou, Henan, P.R. China

<sup>3</sup> Open Access. © 2022 the author(s), published by De Gruyter. © BY This work is licensed under the Creative Commons Attribution 4.0 International License.

of Geographic Information System [2–4], Spatial Database [5–7], and Artificial Intelligence [8–10]. Several problems in reasoning with direction relations have been studied so far, e.g., composition [10–15], inversion [16–19], and consistence checking [20–23] with cardinal direction relations, which play an important role in spatial query, spatial analysis, and planning decision.

In this work, we focus on the inversion operation for cardinal direction relations. The inverse operation for direction relations allows to infer the direction of object *B* with respect to *A*, given the direction of object *A* with respect to *B*, which plays an important role in saving storage and improving the efficiency of spatial queries and spatial reasoning [18]. For example, in an intelligent transportation system, the results of inverse operation for direction relations can be used as the constraints for path selection to reduce the search space and then improve the query efficiency, which can provide supports for choosing reasonable travel or rescue routes when traffic jams or accidents occur. Therefore, it is important to develop the automatic reasoning techniques for the inversion operation of direction relations.

In our work, we aim to solve the problem of automatically computing the inverse of cardinal direction relations in the model of Goyal and Egenhofer [24,25]. This model is currently one of the most expressive models for cardinal directions between regions because it is simple in calculation and it is easy to perform formal reasoning. The specific contributions of this work are summarized as follows.

- We introduce the operations of direction matrix to study the inversion operation for cardinal direction relations in the model of Goyal. We give the representation of the *rectangular* cardinal direction relations in the form of interval matrix. We get the results of the inversion operation of interval matrix by using the inference rules of interval algebra.
- Then, we give several important theorems which provide the solution for the target problem by using the operations of direction matrix, and their proofs to build up our theoretical framework.
- By our theoretical framework, we study the problem of deciding whether a Boolean matrix is geometrically realized or not and the problem of computing all possible original matrices of the given rectangular direction matrix. Then, we give the algorithmic solutions for the two problems.
- Then, we propose our algorithm to compute all possible results of the inverse of the basic cardinal direction relations directly. Several theorems are provided to prove that our algorithms are correct.
- We implement our algorithm in C programs. The results of running the programs show that our algorithm is correct and complete. In this work, we realized the automatic calculation of the inverse of the basic cardinal direction relations in the model of Goyal. Our study makes that model of Goyal has the power of automated reasoning for the inverse operation because our solution does not require the help of reference tables and any manual calculation.

This article is structured as follows. In Section 2, we survey related work. Section 3 presents the direction relation matrix model. In Sections 4 and 5, we introduce the rectangle algebra, and build the inference mechanism of the inverse operation for interval matrix and the correlations between interval matrix and rectangular direction matrix. Section 6 presents our algorithms for the inverse operation of basic cardinal direction relations and proves their correctness. Section 7 discusses the implementation of the proposed algorithms and presents the results of analysis and verification. Finally, Section 8 gives the conclusion and discusses future work.

# 2 Related work

As the basis of the spatial cognition and reasoning with directions, the formal description of spatial directions has received a lot of attention in recent years. Many models for cardinal directions have been proposed in the literature [2,5,24–29]. The model of Goyal and Egenhofer [24,25] is currently one of the most expressive models for cardinal directions in two dimensional space because it can more realistically express the direction relations between regions, and it is simple in calculation and easy to perform formal reasoning. This model is widely used in many fields related to the cognition of spatial direction. For example,

the direction information represented by the model of Goyal can be used in the point of interest recommendation for more effective recommendation. To be specific, it is easy to get interested restaurants in a certain direction by means of this model. Research have been carried out on this interesting model. Goyal and Egenhofer [25] first studied the composition operation on this model and gave a method for computing the consistency-based composition. Skiadopoulos and Koubarakis [10] pointed out that the above composition method of Goyal cannot work correctly for some cases and revised this method, but the corrected method is relatively complicated and it requires the use of composition tables.

Spiros and Manolis [20] also studied the problem of checking the consistency of cardinal directions, and proposed the first algorithm for this problem on the model of Goyal which can be performed in  $O(n^5)$ . Navarrete et al. [22] proposed an  $O(n^4)$  algorithm to check the consistency of the basic cardinal directions between regions in the model of Goyal. As a matter of fact, the consistency checking of a set of unrestricted cardinal directions is a NP-hard problem [23], which is difficult to be solved. Most methods solve this problem on the model of Goyal by using different composition methods based on the traditional path-consistency method, which are generally computationally expensive and inefficient.

In a previous line of work, Liu and Hao [21,30] proposed algorithms for the problems of composition, inversion, and consistency checking with cardinal directions involving only *rectangular* cardinal directions which is an important class of direction relations in the model of Goyal. The above works can only solve these problems with the condition regions should be rectangles or are approximately substituted by their minimum bounding box which may be too crude and thus reduces the precision of expression and reasoning. With the above problem in mind, Chen et al. [12] and Wang et al. [17] extended these methods presented in ref. [30] to deal with the composition and inversion operations for the general case based on the concepts of rectangular cardinal direction and its original relation. However, the above methods still require manual calculations or the use of reference tables in computing the possible original relations of rectangular cardinal directions.

In addition, several improved models [7,31–33] based on the model of Goyal have been proposed by researchers. In these models, the space is further subdivided on the basis of the model of Goyal in different ways. In summary, these models have a finer granularity of space division and can describe the spatial arrangements more accurately. Thus, directions in these models are clearly more expressive than the model of Goyal, but it is difficult to perform formal reasoning and calculation for these models.

In this work, we will focus on the inversion operation of directions which is a fundamental problem in qualitative spatial reasoning. In earlier studies on this problem, the concerned spatial objects mainly involve points, intervals [34], and rectangles [17,21,30]. Obviously, it is necessary to deal with the above problem for general regions. Summarizing, the model of Goyal is a more expressive model of qualitative reasoning with cardinal directions between regions. We will employ this model to handle the inversion operation for cardinal direction relations between regions. Although, there are several works [10,13,17,20] on qualitative reasoning with cardinal directions in this model, the ability of automated reasoning is still very poor. Especially for the inversion operation, the only published method presented in ref. [17] still needs to employ inference tables and manual calculations. In the following sections, we will present an algorithm that automatically performs the inversion operation for cardinal directions in the model of Goyal. This algorithm can be widely used in a lot of areas such as resource allocation, emergency rescue, disaster prevention and mitigation, and intelligent transportation. For example, in an intelligent transportation system, the results of inverse operation computed by our algorithm for direction relations can be used as the constraints for path selection to reduce the search space and then improve the query efficiency, which can provide supports for choosing reasonable travel or rescue routes when traffic jams or accidents occur.

## 3 Direction relation matrix model

In this work, we employ the model introduced by Goyal and Egenhofer [24,25] for cardinal direction relations which are used to describe how one region called primary region is placed relative to another called reference region. The concerned regions are homeomorphic to the closed unit disk ( $\{(x, y) \in R^2 | x^2 + y^2 \le 1\}$ ). Such a set of

these regions will be denoted by *REG*. Briefly, regions in *REG* are closed, connected, and have connected boundaries. Let *a* be a region in *REG*. The projection of *a* on the *x*-axis (respectively *y*-axis) is denoted by  $a_x$  (respectively  $a_y$ ). The greatest lower bound or *infimum* of  $a_x$  (respectively  $a_y$ ) is denoted by  $inf_x(a)$  (respectively  $inf_y(a)$ ). The least upper bound or the *supremum* of  $a_x$  (respectively  $a_y$ ) is denoted by  $sup_x(a)$  (respectively  $sup_y(a)$ ). The minimum bounding box of region *a*, denoted by mbb(a), is the box formed by the straight lines  $x = inf_x(a)$ ,  $x = sup_x(a)$ ,  $y = inf_y(a)$ , and  $y = sup_y(a)$ .

The model of Goyal adopts a projection-based system placed around the reference region. Let region *a* in *REG* be the reference region. Such a system induced by the axes forming mbb(a), divides the plane into nine areas NW(a), N(a), NE(a), W(a), B(a), E(a), SW(a), S(a), and SE(a) which we call direction *tiles* of *a* (Figure 1).

If the primary region *b* in *REG* is included in tile N(a), then we say that *b* is north of *a* and we write *b* N *a*. Similarly, we can define northwest (NW), northeast (NE), west (W), bounding box (B), southwest (SW), east (E) and southeast (SE) relations. The above relations called single-tile cardinal direction relations are defined formally as following:

*b* NW *a* iff  $sup_x(b) \le inf_x(a)$  and  $sup_y(a) \le inf_y(b)$  *b* N *a* iff  $sup_y(a) \le inf_y(b)$ ,  $inf_y(a) \le inf_x(b)$ , and  $sup_x(b) \le sup_y(a)$  *b* NE *a* iff  $sup_x(b) \le inf_x(a)$  and  $sup_y(a) \le inf_y(b)$  *b* W *a* iff  $sup_x(b) \le inf_x(a)$ ,  $inf_y(a) \le inf_x(b)$ , and  $sup_y(b) \le sup_y(a)$  *b* B *a* iff  $inf_x(a) \le inf_x(b)$ ,  $sup_x(b) \le sup_x(a)$ ,  $inf_y(a) \le inf_y(b)$ , and  $sup_y(b) \le sup_y(a)$  *b* E *a* iff  $sup_x(a) \le inf_x(b)$ ,  $inf_y(a) \le inf_x(b)$ , and  $sup_y(b) \le sup_y(a)$  *b* SW *a* iff  $sup_x(b) \le inf_x(a)$  and  $sup_y(b) \le inf_y(a)$  *b* S *a* iff  $sup_y(b) \le inf_y(a)$ ,  $inf_x(a) \le inf_x(b)$ , and  $sup_x(b) \le sup_y(a)$  *b* SE *a* iff  $sup_x(a) \le inf_y(a)$ ,  $inf_x(a) \le inf_x(b)$ , and  $sup_x(b) \le sup_y(a)$ *b* SE *a* iff  $sup_x(a) \le inf_y(a)$ ,  $inf_x(a) \le inf_y(a)$ 

However, the primary region b may fall into more than one tile. For instance the primary region b lies partly in the *tile* NW(a) and partly in the *tile* W(a) of a (Figure 1), then we say that b is partly northwest and partly east of a and we write a NW:W b.

Without loss of generality, the definition of basic cardinal direction relation is provided as following on the basis of the single-tile cardinal direction relations.

**Definition 1.** A basic cardinal direction relation is an expression  $R_1:...:R_k$  where  $R_1,...,R_k \in \{B, S, SW, W, NW, N, NE, E, SE\}$ ,  $1 \le k \le 9$ , and  $R_i \ne R_j$  for every i, j such that  $1 \le i, j \le k$  and  $i \ne j$ , and there exist regions  $b_1,...,b_k \in REG$  such that  $b_1R_1$   $a,...,b_k R_k$  a and  $b_1 \cup \cdots \cup b_k \in REG$  for any reference region  $a \in REG$ .

The set of basic cardinal direction relations that contains 218 elements is denoted by *D*. Then, a cardinal direction relation is defined as an element of the power set  $2^{D}$  of *D*. A cardinal direction relation can be seen as the union of basic cardinal direction relations, which can be used to describe not only definite but also indefinite information about cardinal directions, e.g., *b* {*NW*,*E*} *a* means that *b* is northwest or east of *a*.

$$\operatorname{dir}(b, a) = \begin{bmatrix} NW(a) \cap b & N(a) \cap b & NE(a) \cap b \\ W(a) \cap b & B(a) \cap b & E(a) \cap b \\ SW(a) \cap b & S(a) \cap b & SE(a) \cap b \end{bmatrix}.$$
(1)



In this model, the cardinal direction relation of the primary region *b* relative to the reference region *a* is stored in a Boolean  $3 \times 3$  matrix (equation (1)) called the direction relation matrix. Such matrix is generated by checking whether the intersection of the primary object and the tiles of the reference object is empty or not. For each tile, if such intersection is non-empty, it means that the primary object falls into this direction tile. In this work, for convenience, if such intersection is empty, it is recorded by 0, else recorded by 1 in such matrix. For instance, as shown in Figure 1, the direction *c* with respect to *a* can be represented by the direction matrix  $M_1$  and we write  $c P_1 b$ .

$$P_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$
 (2)

**Definition 2.** A basic cardinal direction relation *R* is called *rectangular*, if there exist two rectangles (with sides parallel to the *x*- and *y*-axes) *a* and *b* such that *a R b* is satisfied; otherwise, it is called *non-rectangular*.

**Definition 3.** A basic cardinal direction relation matrix *P* is called *rectangular* if all the non-zero elements in *P* form a rectangle; otherwise, it is called *non-rectangular*.

By Definition 2, we have that there are 36 *rectangular* cardinal direction relations as follows: *NW*, *N*, *NE*, *W*, *B*, *E*, *SW*, *S*, *SE*, *NW*:*N*, *N*:*NE*, *NW*:*W*, *N*:*B*, *NE*:*E*, *W*:*B*, *B*:*E*, *W*:*SW*, *B*:*S*, *E*:*SE*, *SW*:*S*, *S*:*SE*, *NW*:*N*:*NE*, *W*:*B*:*E*, *SW*:*SSE*, *NW*:*W*:*SW*, *N*:*B*:*S*, *NE*:*E*:*SE*, *NW*:*N*:*W*:*B*, *N*:*NE*:*B*:*E*, *W*:*B*:*SW*:*S*, *B*:*E*:*SSE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*SW*:*S*, *B*:*E*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*SW*:*S*, *B*:*E*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*S*, *NE*:*E*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*SW*:*S*, *B*:*E*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*S*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*S*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*S*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*, *W*:*B*:*S*:*SE*, *NW*:*N*:*NE*:*W*:*B*:*E*:*SW*:*S*:*SE*, *N*:*SW*:*S*:*SE*, *N*:*SW*:*S*, *S*:*SE*, *N*:*SW*:*S*:*SE*, *N*:*SW*:*S*, *S*:*SE*, *N*:*SW*:*S*:*S*, *S*:*SE*, *N*:*SW*:*SW* 

**Example 1.** The direction matrix  $P_2$  corresponding to the *rectangular* cardinal direction relation *NW*:*N*:*W*:*B* is *rectangular*, while  $P_3$  is *non-rectangular* for the *non-rectangular* relation *N*:*N*:*E*:*S*:*S*.

$$P_2 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},\tag{3}$$

$$P_3 = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$
 (4)

In Sections 4 and 5, we aim to solve the problem of computing the inverse of basic cardinal direction relations which is defined formally as following.

**Definition 4.** Let  $R \in 2^{D}$ . The *inverse* of relation *R*, denoted by *INV*(*R*), is another cardinal direction relation which satisfies the following. For arbitrary regions *a*, *b*  $\in$  *REG*, *a R b* holds, iff *b INV*(*R*) *a* holds.

## 4 Rectangle algebra and rectangular cardinal direction relation

Interval algebra was introduced by Allen [34] for temporal reasoning. There are 13 basic relations between two temporal intervals which form a set of jointly exhaustive and pairwise disjoint relations (Table 1). The set of symbols  $A_{int} = \{p,m,o,s,d,f,pi,mi,oi,si,di,f,eq\}$  is used to denote the 13 basic relations.

Similar to cardinal direction relation matrix, we can also use a Boolean matrix called interval relation matrix to describe the relations between the intervals. The one dimensional space is divided into three parts by the reference interval a (Figure 2). This matrix records into which parts the primary interval b falls by checking whether the intersection of the primary interval b and each part is empty or not. As shown in

#### Table 1: Interval algebra relations

Interval Algebra (r)	Constraints		Inverse $(r^{-1})$
р	$sup(b) \le inf(a)$		pi
т	sup(b)=inf(a)	├─── <b>├</b> ─ ─ ─ ─	mi
0	$inf(b) \le inf(a) \land sup(a) \le sup(b)$	┝──────────	oi
S	$inf(a)=inf(b) \land sup(b) \leq sup(a)$	<b>⊢−−</b> ′−ı	si
d	$inf(a) \leq inf(b) \land sup(b) \leq sup(a)$	⊢ <sup> </sup> <u> </u>	di
f	$inf(a) \le inf(b) \land sup(a) = sup(b)$	⊢ – <u>⊢ – –</u> 1	fi
eq	$inf(a)=inf(b) \land sup(a)=sup(b)$	<b>⊨</b> = = =1	eq

Figure 2: Example of interval relation matrix.

Figure 2, the relation of *b* with respect to *a* can be represented by interval relation matrix [110] and we write b[110]a.

There are six basic interval relation matrices defined as follows:

b [1 0 0] a iff  $sup(b) \le inf(a)$ .

b [1 1 0] a iff  $inf(b) < inf(a) < sup(b) \le sup(a)$ .

b [1 1 1] a iff inf(b) < inf(a) and sup(a) < sup(b).

- b [0 1 0] a iff  $inf(a) \le inf(b)$  and  $sup(b) \le sup(a)$ .
- b [0 1 1] a iff  $inf(a) \le inf(b)$  and sup(a) < sup(b).
- b [0 0 1] a iff  $sup(a) \leq inf(b)$ .

According to the definitions of interval relation matrix and interval algebra relations, there exists a mapping between interval relation matrix and interval algebra relations as shown in Table 2. For each pair of mapping, the corresponding interval relation matrix and interval algebra relation have the same spatial constraints between the intervals.

Rectangular algebra proposed by Balbiani et al. [35,36] is the extension of interval algebra to the bidimensional space. It describes the spatial configurations between rectangles whose sides are parallel to the axes of some orthogonal basis in a bi-dimensional Euclidean space. A basic rectangular algebra relation is defined as an element of the set  $A_{rec} = \{(R_x, R_y) | R_x, R_y \in A_{int}\}$ . For each relation  $(R_x, R_y) \in A_{rec}$ , there exist two rectangles *a* and *b* such that  $b(R_x, R_y) a$  holds iff  $b_x R_x a_x$  and  $b_y R_y a_y$  hold.

Table 2: The mapping between interval relation matrices and interval algebra relations

Interval matrix	Constraints	Interval algebra					
[1 0 0]	$sup(b) \leq inf(a)$	{ <b>p</b> , <b>m</b> }					
[1 1 0]	$inf(b) < inf(a) < sup(b) \le sup(a)$	{ <b>o</b> ,fi}					
[1 1 1]	$inf(b) < inf(a) \land sup(a) < sup(b)$	{ <i>di</i> }					
[0 1 0]	$inf(a) \leq inf(b) \wedge sup(b) \leq sup(a)$	{ <i>d</i> , <i>s</i> , <i>f</i> , <i>e</i> }					
[0 1 1]	$inf(a) \le inf(b) \land sup(a) < sup(b)$	{ <i>si,oi</i> }					
[0 0 1]	$sup(a) \leq inf(b)$	{ <b>pi,mi</b> }					



Figure 3: Illustration of Example 2.

**Example 2.** As shown in Figure 3, there exist rectangles *a* and *b* such that b(p,m) a holds. We can also see that  $b_x p a_x$  and  $b_y m a_y$  hold.

Then, a rectangular algebra relation is defined as an element of the power set  $2^{Arec}$  of  $A_{rec}$ . It can be used to describe not only definite but also indefinite information about spatial configurations between rectangles. If there exist rectangles *a* and *b* such that *b R a* is satisfied, where *R* is a rectangular algebra relation, if and only if one of the basic relation in *R* is satisfied, e.g.,  $b\{(pi,f), (d,e)\}$  *a* denotes that *b* (pi,f) *a* or *b* (d,e) *a* for rectangles *a* and *b*.

Both *rectangular* cardinal direction relation and rectangular algebra can be used to describe the spatial configurations between rectangles in *REG*. Let us consider two rectangles *a* and *b* and assume that *b R a*, where *R* is a basic rectangular cardinal direction relation, then there exists a rectangular algebra relation *r* such that *b r a* holds, and *vice versa*. Thus, there must be a mapping between *rectangular* cardinal direction and rectangular algebra relations. Let us consider the basic rectangular cardinal direction relation *SW:S* and assume that there exist rectangles *a* and *b* such that *b SW:S a* holds. According to the definition of basic cardinal direction relation (Definition 1), we have *b SW:S a* if and only if  $inf_x(b) < inf_x(a) < sup_x(b) \le sup_x(a)$  and  $sup_y(b) \le inf_y(a)$ , from which we have  $b_x$  {o, *fi*}  $a_x$  and  $b_y$  {*p,m*}  $a_y$  according to the definition of interval algebra relations presented in Table 1, namely, *b* {*o, fi*} × {*p,m*} *a*. Thus, rectangular cardinal direction relation *SW:S* and rectangular algebra relation {*o, fi*} × {*p,m*} are a pair of mapping. In similar way, we build a mapping between *rectangular* cardinal directions and rectangular algebra relations which is presented in the form of matrix as shown in Table 3.

Y/X	<b>[1 0 0]</b>	<b>[1 1 0]</b>	[1 1 1]	[0 1 0]	[0 1 1]	[0 0 1]
$\begin{bmatrix} 1\\0\\0\end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 1\\1\\0\end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 1\\1\\1\\1\end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$
$\begin{bmatrix} 0\\1\\0\end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$
$\begin{bmatrix} 0\\1\\1\end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$
$\begin{bmatrix} 0\\0\\1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Table 3: The mapping between rectangular cardinal direction relations and rectangular algebra relations

# 5 Correlations between interval relation matrix and rectangular cardinal direction matrix

There exist equivalent correlations between interval relation matrix and interval algebra relations, so that we can get the inverse matrix by means of the inference rules of interval algebra for basic interval relation matrix. For interval relation matrix [1 0 0], we have its equivalent interval algebra relation  $\{p,m\}$  from Table 3. Thus, the interval algebra corresponding to  $INV([1 \ 0 \ 0])$  is  $\{p,m\}^{-1}$  which equals  $\{pi,mi\}$ . From Table 2, we have  $\{pi,mi\}$  is equivalent to  $[0 \ 0 \ 1]$ , so we have  $INV([1 \ 0 \ 0]) = [0 \ 0 \ 1]$ . In the same way, we have the inverse of basic interval relation matrices presented in Table 4.

**Definition 5.** Let *P* be a basic cardinal direction relation matrix, the projection of *P* on the *x*-axis, denoted by  $P_x$  is a 1 × 3 matrix, if there exists at least one P(i,j) = 1 for each  $I \in \{1,2,3\}$ , then  $P_x(1,j) = 1$ , else  $P_x(1,j) = 0$ ,  $j \in \{1,2,3\}.$ 

**Definition 6.** Let *P* be a basic cardinal direction relation matrix, the projection of *P* on the *y*-axis, denoted by  $P_x$ , is a 3 × 1 matrix, if there exists at least one P(i,j) = 1 for each  $j \in \{1,2,3\}$ , then  $P_v(i,1) = 1$ , else  $P_x(i,1) = 0$ ,  $I \in \{1, 2, 3\}.$ 

**Example 3.** For direction matrix  $P = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$ , we have  $P_x = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$  and  $P_y = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ .

**Lemma 1.** Let P be a rectangular direction matrix, the rectangular algebra  $(P_x, P_y)$  presented by interval relation matrix is the equivalent relation of P.

Proof. This proof is given by case analysis. There are 36 basic rectangular direction matrices. Let us fist consider that  $P = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ . Then, we have  $P_x = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}$  and  $P_y = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ . We have that the equivalent rectangular algebra of *P* represented by interval relation matrix is  $\begin{pmatrix} 1 & 1 & 0 \end{bmatrix}$ ,  $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ , namely,  $(P_x, P_y)$ , thus

the lemma holds for this case. Similarly, we have the lemma hold for any other basic rectangular direction matrix and thus the lemma holds. п

**Lemma 2.** Let P be a rectangular direction matrix, then  $P = P_v \times P_x$ .

**Proof.** This proof is given by case analysis. There are 36 basic *rectangular* direction matrices. Let us fist consider that  $P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ . Then we have  $P_x = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$  and  $P_y = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ . According to the definition of matrix

Interval matrix ( <i>M</i> )	Inverse ( <i>INV</i> ( <i>M</i> ))
[1 0 0]	[0 0 1]
[1 1 0]	[0 1 0], [0 1 1]
[1 1 1]	[0 1 0]
[0 1 0]	[0 1 0], [0 1 1], [1 1 0], [1 1 1]
[0 1 1]	[0 1 0], [1 1 0]
[0 0 1]	[1 0 0]

Table 4: Inverse of interval matrix

multiplication, we have  $\begin{bmatrix} 1\\1\\0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0\\0 & 1 & 0\\0 & 0 & 0 \end{bmatrix}$ . Thus, the lemma holds for this case. Similarly, the lemma holds for any other basic *rectangular* direction matrix and thus the theorem holds.

# 6 Computing the inverse of basic cardinal direction relations

In this Section, we will focus on the problem of computing the inverse of basic cardinal direction relations. We will need the following definitions and theorems.

**Definition 7.** Let *P*, *Q* be two basic cardinal direction matrices. For each  $i,j \in \{1,2,3\}$ , if  $Q_{ij} = 1$ , then we must have  $P_{ii} = 1$ , then we say that *P* includes *Q*.

**Definition 8.** Let *P* be a basic direction matrix. The *bounding matrix* of *P*, denoted by Br(R) is the smallest rectangular direction matrix (with respect to the number of non-zero elements) that includes *P*.

**Definition 9.** Let *P* be a rectangular direction matrix, the original matrix of *P* is a basic direction matrix with that its bounding matrix is *P*. The set of original matrices of *P* is denoted by ORG(P).

	1	1	1		1	1	1		1	1	1		1	1	1
Example 4.	1	1	1	includes	1	1	0	, and	1	1	0	is the original matrix of	1	1	1
	0	0	0		0	0	0_		0	0	0		0	0	0

**Lemma 3.** Let *P* be a basic direction matrix, and *a*, *b* be regions in REG such that *a P b* holds, then mbb(*a*) Br (*P*) *b* also holds.

**Proof.** It is easy to see that if a P b holds, then mbb(a) Br(P) b also holds.

For	instance,	if	а	$\begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$	1 1 0	b,	then	mbb(a)	0 0 0	1 1 0	1 1 0	b	(Figure	4),	where
$Br\left(\left[\begin{array}{rrr} 0 & 1\\ 0 & 0\\ 0 & 0 \end{array}\right]\right)$	$\begin{bmatrix} 1\\1\\0 \end{bmatrix} = \begin{bmatrix} 0\\0\\0 \end{bmatrix}$	1 1 1 1 0 0	L] L]. )]												

**Theorem 1.** Let *P* be a rectangular direction matrix,  $INV_{rec}(P)$  be the set of rectangular matrices in the inverse matrices of *P*, then

 $INV_{rec}(P) = \{p \times q | p \in INV(P_v) \land q \in INV(P_x)\}.$ 

**Proof.**  $\forall t \in INV_{rec}(P)$ , there exist two rectangles a, b such that  $a \ t \ b$  and  $b \ P \ a$  hold. Then, according to Lemma 1, we have  $a_x \ t_x \ b_x \land a_y \ t_y \ b_y \land b_x P_x a_x \land b_y P_y a_y$ . Thus, we have  $t_x \in INV(P_x) \land t_y \in INV(P_y)$ . Then, according to Lemma 2, we have  $t = t_y \times t_x$ .

Therefore, we have  $t \in \{p \times q | p \in INV(P_v) \land q \in INV(P_x)\}$ .



Figure 4: Proving Lemma 3.

Conversely,  $\forall t \in p \times q | p \in INV(P_y) \land q \in INV(P_x)$ }, according to Lemma 2, we have  $t = t_y \times t_x$ , and since  $t \in \{p \times q | p \in INV(P_y) \land q \in INV(P_x)\}$ , then,  $t_y \in INV(P_y) \land t_x \in INV(P_x)$ , and then  $t_y \times t_x \in INV_{rec}(P)$ . Since  $t = t_y \times t_x$ , then  $t \in INV_{rec}(P)$  and thus the theorem holds.

**Theorem 2.** Let *Q* be a basic direction matrix, INV(Q) be the set of the inverse matrices of *Q*, then,  $INV(Q) = \{ORG(p \times q) | p \in INV(Br(Q)_v) \land q \in INV(Br(Q)_x)\}$ 

**Proof.** Let us first prove that  $INV(Q) = \{ORG(t) | t \in INV_{rec}(Br(Q))\}$  holds.  $\forall q \in INV(Q)$ , let *a*, *b* be regions in *REG* and *a q b*  $\land$  *b Q a* holds. Then, *mbb*(*a*) Br(q) *b*  $\land$  *mbb*(*b*) Br(Q) *a* holds by Lemma 3. Equivalently, *mbb*(*a*) Br(q) *mbb*(*b*)  $\land$  *mbb*(*b*) Br(Q) *mbb*(*a*) holds. Then, according to the definition of inverse,  $Br(q) \in INV_{rec}(Br(Q))$  holds. Therefore, according to the definition of original matrix (Definition 9),  $q \in \{ORG(t) | t \in INV_{rec}(Br(Q))\}$  holds.

Conversely,  $\forall t \in INV_{rec}(Br(Q))$ , there exist rectangles *a* and *b* such that *a*  $t b \land b Br(Q) a$  holds.  $\forall q \in ORG$ (*t*), assume that  $r_1:...:r_k$  is the cardinal direction corresponding to *q*, assume also that  $R_1:...:R_j$  is the cardinal direction corresponding to *Q*, let us form region  $b_0 = b \cap (R_1(a) \cup ... \cup R_j(a))$  and region  $a_0 = a \cap (r_1(b) \cup ... \cup r_k(b))$ . By the construction of regions  $a_0$  and  $b_0$ , have  $mbb(a_0) = a$  and  $mbb(b_0) = b$ . Then, we have  $b_0 R_1:...:R_j$   $a_0 \land a_0 r_1:...:r_k b_0$  holds by the construction of regions  $a_0$  and  $b_0$  and the definition of basic cardinal direction relation (Definition 1), namely,  $b_0 Q a_0 \land a_0 q b_0$  holds.

For instance, if  $aBr(N:NE:E)b \wedge bBr(W:SW:S) a$  holds (Figure 5(a)), then  $a_0 N:NE:E b_0 \wedge b_0 W:SW:S a_0$  also holds, where  $a_0 = a \cap (N(b) \cup NE(b) \cup E(b))$  and  $b_0 = b \cap (W(a) \cup SW(a) \cup S(a))$  (Figure 5(b)).

Since  $b_0 Q a_0 \wedge a_0 q b_0$  holds, then we have  $q \in INV(Q)$  holds by Definition 4. Thus,  $INV(Q) = \{ORG(t) | t \in INV_{rec}(Br(Q))\}$  holds. We have  $INV_{rec}(Br(Q)) = \{p \times q | p \in INV(Br(Q)_y) \land q \in INV(Br(Q)_x)\}$  by Theorem 1, and thus the Theorem holds.

The above theorem offers a method to compute the inverse of basic cardinal direction relation. By this theorem, we have to get the set ORG(P) for a rectangular direction matrix *P*. The set ORG(P) can be obtained by manual reasoning and calculating. First, the invalid direction matrices are eliminated from the set of matrices included by *P*. Then, all the possible original matrices of *P* will be found by manual checking according to Definition 9.

We illustrate the above manual operation with the following example.

**Example 5.** Let P be
  $\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ . There are fifteen matrices included by P, as follows:

  $\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$   $\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ 

we cannot find out the two regions *a*,*b* in *REG* such that *a M b* holds, so that *M* is invalid by Definition 9. The other matrices are direction matrices.



Figure 5: Proving Theorem 2.

 $\forall M \in \left( \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)$ Br(M) = M, so that the eight direction matrices are not the original matrices of P. $ORG(P) = \left( \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, we have Br(M) = P. Then, according to Definition 9, we have$  $<math display="block"> \left( \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)$ 

 $ORG(P) = \left( \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right)$ 

Notice that the above process of manual reasoning is very complicated as it needs to check all the matrices included by the rectangular direction matrix *P* to find out the target matrices. By Definition 7, there are  $2^n - 1$  matrices included by a given rectangular direction matrix *P* with  $n(1 \le n \le 9)$  non-zero elements. Therefore, the amount of work on manual reasoning increases sharply as the number of non-zero elements in *P* grows. So, it is impractical to employ manual reasoning. Therefore, to solve above problem the following problem should be solved.

**Problem 1.** Given an arbitrary rectangular direction matrix *P*, *ORG*(*P*) is calculated automatically.

By Definition 9, if matrix *M* in the set of the Boolean  $3 \times 3$  matrices is a valid direction matrix such that Br(M) = P holds, then  $M \in ORG(P)$  holds, for a rectangular direction matrix *P*. Thus, we can get ORG(P) by checking each matrix *M* in the set of the Boolean  $3 \times 3$  matrices. Therefore, to solve problem 1, we first have to solve the following problem.

**Problem 2.** Given an arbitrary Boolean  $3 \times 3$  matrix *M*, decide whether *M* is a valid direction matrix or not.

The following Lemma provides necessary and sufficient conditions to determine the validity of a given matrix.

**Lemma 4.** Given an arbitrary Boolean 3 × 3 matrix *M*, *M* is a valid direction matrix if and only if it is non-empty and 4-connected.

We provide an approach to determine whether a given matrix is 4-connected or not by using auxiliary queue in the following way.

Initially, start with any non-zero element in matrix *M*, record its row and column number into the queue and then set it to be zero. Then, repeat the following process we call 4-neighbors, search when the queue is non-empty. Remove an element from the front of the queue and then, for each non-zero element in the four neighbors which are horizontally or vertically adjacent to the removed element, add its row and column number to the queue and set it to be zero. When the queue is empty, if there does not exist non-zero element in *M*, then *M* is 4-connected, else it is not.

By using the above approach and Lemma 4, we get the following algorithm to solve problem 2.

## Algorithm. VALID-CDM

Input: An arbitrary Boolean 3×3 matrix *M* Output: true or false begin int *row*[4] = {0,0,-1,1};int *col*[4] = {1,-1,0,0}; struct *Point* {int *x*,*y*;}; *queue < Point > que*;

```
int cnt: = 0; bool flag: = false;
  for int i: = 1 to 3 do
     for int j: = 1 to 3 do
       if(M[i][j] = 1), then
         M[i][j] = 0; flag = true; cnt ++;
         que.push(Point{i,j});
         while(!que.empty()) do
            {Point q = que.front();}
               que.pop();
                 for int k: = 0 to 3 do
                         \{ int \ x = q.x + row[k]; int \ y = q.y + col[k]; \}
                                     if (x > = 1\&\&x < = 3\&\&y > = 1\&\&y < = 3\&\&M[x][y] = 1) then
                                     M[x][y]: = 0; que.push(Point{x,y});
                                     end if
    end if
  if (cnt > 1 or flag = false) then return false;
    else return true;
  end if
end
```

**Theorem 3.** Algorithm VALID-CDM is correct, i.e., it returns whether a Boolean  $3 \times 3$  matrix *M* is a valid direction matrix or not.

**Proof.** This algorithm adopts the non-zero element encountered first in executing the for loop as the starting element. Then, the iterative 4-neighbors search that starts with this non-zero element is performed by executing the inner while loop. When the queue becomes empty and then the while loop terminates, if there does not exist non-zero element in *M*, then the for loop terminates with *cnt* = 1 and this algorithm ultimately returns true. For this case, we have the fact that all the non-zero elements in *M* can be found by once iterative four-neighbors search which indicates *M* is 4-connected. By Lemma 4, we have that *M* is a valid direction matrix. Therefore, the result of this algorithm is correct for this case.

Otherwise, the for loop terminates when  $cnt > 1 \lor flag$  = false holds and then this algorithm returns false. If flag = false, which implies *M* is empty, then *M* is invalid by Lemma 4. If cnt > 1 which indicates the non-zero elements of *M* cannot be found by once iterative four-neighbors search, then we can conclude that the primary object corresponding to *M* consists of at least two regions in *REG*, which contradicts with the definition of basic cardinal direction relation (Definition1). Therefore, the result of this algorithm is correct and thus this theorem holds

We give an example of Algorithm VALID-CDM in operation as following.

**Example 6.** Given two Boolean  $3 \times 3$  matrices  $M_1$  and  $M_2$  (Figure 6), it is easy to see that  $M_1$  is a valid direction matrix, while  $M_2$  is not. By using Algorithm VALID-CDM to examine the above two matrices, we can see that there exists non-zero element  $M_2[2][1]$  in  $M_2$ , while it does not in  $M_1$  when the query is empty (Figure 6). Thus, the results of Algorithm VALID-CDM are in agreement with the actual states.

In the following, we will combine Algorithm VALID-CDM to solve Problem 1. We know that the set of Boolean  $3 \times 3$  matrices can be represented by a full binary tree with depth 9. Thus, the original matrices of a rectangular direction matrix can be computed by implementing depth-first search in such binary tree. During this search, a Boolean  $3 \times 3$  matrix is obtained whenever a leaf node is reached. For each obtained matrix, we check whether it is a direction matrix or not by calling algorithm VALID-CDM. If it is valid, then we check whether it is the original matrix of the given rectangular matrix or not. By the above method, an algorithm for computing all the original matrices of a given rectangular direction matrix is presented as follows.



Figure 6: Illustration of Algorithm VALID-CDM in operation.

## Algorithm. COMPUTE-ORG

```
Input: An arbitrary rectangular direction matrix P, integer a
Output: ORG(P)
begin
    int i, j;
    if (a > 9)
         q:=p;
         if (VALID-CDM(q) \text{ and } Br(p) = P), then
           R: = R \cup \{p\};
            return;
         end if
    end if
  if (a \mod 3 = 0), then
         i = a/3; j = 3;
  else
       i = (a/3) + 1; j = a \mod 3;
  end if
  p[i][j]: = 1;
  COMPUTE-ORG(P, a + 1);
  p[i][j]:=0;
  COMPUTE-ORG(P, a + 1);
end
```

**Theorem 4.** Algorithm COMPUTE-ORG is correct and complete, i.e., it returns all the possible original matrices of a rectangular direction matrix P.

**Proof.** This algorithm performs a naive search in the set of Boolean  $3 \times 3$  matrices by using recursive call. During the execution, a Boolean  $3 \times 3$  matrix p is obtained whenever the integer a becomes 10. Then, it invokes Algorithm VALID-CDM to decide whether p is a valid cardinal direction matrix or not. The correctness of Algorithm VALID-CDM has been proved in Theorem 3. If p is a valid direction matrix and Br(p) = P also holds, then p is recorded into R. We have that p is the original matrix of P by Definition 9. Thus, R ultimately records ORG(P) as there is no omission by this naive search. Therefore, this algorithm is correct and complete.

In the remainder of this section, we will give an algorithm for computing the inverse of the basic cardinal direction relation.

By Theorem 2, the inverse of a basic direction matrix *P* can be computed by the following way. The rectangular matrix Br(P) is calculated first. Then, Br(P) is projected to the *x*-axis and *y*-axis, respectively. Furthermore, we take the inverse operation for the projection matrix of Br(P) on *x*-axis and *y*-axis, respectively. Then, we get the set  $INV(Br(P)_y) \times INV(Br(P)_x)$ . Finally, we employ algorithm COMPUTE-ORG to compute the original matrices for each matrix in  $INV(Br(P)_y) \times INV(Br(P)_x)$ . Notice that the set of the original matrices obtained is the result of the inversion operation of *P*. Based on the above discussion, we present our algorithm as following.

#### Algorithm. COMPUTE-INV

input: An arbitrary basic direction matrix P Output: inverse of P begin Q = Br(P);for *i*: = 1 to 3 do for j = 1 to 3 do if(Q[i][j] = 1) then {*Y*[*i*] = 1; break;} for i = 1 to 3 do for i = 1 to 3 do if(Q[j][i] = 1), then  ${X[i] = 1; break;}$ R := INV(Y);S := INV(X); $C := \emptyset$ for each  $r \in R$ for each  $s \in S$  $C = C \cup \text{Computer-ORG}(r \times s, 1);$ return *C*; end

**Theorem 5.** Algorithm COMPUTE-INV is correct and complete, i.e., it returns the inverse of a basic cardinal *direction matrix P.* 

**Proof.** It is easy to see that this algorithm follows the approach presented in Theorem 2 and invokes Algorithm COMPUTE-ORG. Thus, the correctness and completeness of this algorithm follows from the correctness and completes of Algorithm COMPUTE-ORG and the correctness of Theorem 2 which have been proved.

# 7 Algorithm analysis and discussion

In Section 6, we have presented algorithm COMPUTE-INV that computes the inverse of basic cardinal direction relations. In this section, we will verify the correctness and completeness of the proposed algorithm by comparing the result of this algorithm with that of manual reasoning for each basic cardinal direction relation.

We have implemented the Algorithm VALID-CDM, COMPUTE-ORG, and COMPUTE-INV in C programming language that runs on Visual C++ 6.0. We employ two dimensional arrays to store direction relation matrices. The queue used in Algorithm VALID-CDM with element  $\langle i,j \rangle$  (*i* and *j* are two integers) is implemented by using the C++ Standard Template Library. Algorithm COMPUTE-ORG performs the depth-first search in the set of Boolean  $3 \times 3$  matrices which is implemented by recursive programming. Although the set of Boolean  $3 \times 3$ matrices can be seen as a full binary tree with depth 9, there is no need to use the data structure binary tree as such binary tree can be simulated by each element in the direction matrix set to be 0 or 1 randomly.

The inverse of each basic direction matrix has been generated by Algorithm COMPUTE-INV and manual reasoning, respectively. The results and the code are available from us. Then, we will check whether the result of Algorithm COMPUTE-INV is consistent with that of manual reasoning for each basic cardinal direction relation.

An example of the inverse of the basic direction matrix generated by Algorithm COMPUTE-INV is presented as follows.

For instance, we want to calculate the basic direction matrix  $M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ . *Br*(M) is calculated first.

We have  $Br(M) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ .

So, the projection of Br(M) on the *y*-axis is  $\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$ , and the projection of Br(M) on the *x*-axis is  $\begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$ .

Then, we take the inverse operation for the projection matrix of Br(Q) on x-axis and y-axis, respectively.

We have  $INV\left(\begin{bmatrix}0\\1\\1\end{bmatrix}\right) = \left(\begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}1\\1\\0\end{bmatrix}\right)$  and  $INV(\begin{bmatrix}0 \ 1 \ 1\end{bmatrix}) = (\begin{bmatrix}0 \ 1 \ 0\end{bmatrix}, \begin{bmatrix}1 \ 1 \ 0\end{bmatrix})$ . Therefore, we have

$$INV(M) = ORG\left(\begin{bmatrix} 0\\1\\0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}\right) \cup ORG\left(\begin{bmatrix} 0\\1\\0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}\right) \cup ORG\left(\begin{bmatrix} 1\\1\\0 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}\right)$$
$$\cup ORG\left(\begin{bmatrix} 1\\1\\0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}\right)$$

Equivalently, we have

$$INV(M) = ORG\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) \cup ORG\left(\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) \cup ORG\left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) \cup ORG\left(\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)$$

We know that

$$ORG\left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) = \left\{ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\}, ORG\left(\begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) = \left\{ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\}, ORG\left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) = \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\}$$
$$ORG\left(\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) = \left\{ \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right\}$$

Therefore, we have

$$INV\left( \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \right) = \left\{ \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1$$

Then, the inverse of the basic direction matrix is generated by manual reasoning, shown as follows.  $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ 

Assume that *a* and *b* are regions in *REG* such that  $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$  *b* holds (Figure 7). We have the possible

spatial configurations of b with respect to a (also see Figure 7).

## We have

 $b B a \lor b W:B a \lor b N:B a \lor b N:W:B a \lor b NW:W:B a \lor b NW:N:W a \lor b NW:N:B a \lor b NW:N:W:B a$  holds. Equivalently, we have

	ſI	0	0	0		0	0	0]	0	1	0		0	1	0]		1	0	0]	ſ	1	1	0]	Í	1	1	0		1	1	0])
Dir(b, a) = b	łI	0	1	0	,	1	1	0,	0	1	0	,	1	1	0	,	1	1	0	,	1	0	0	,	0	1	0	,	1	1	0
	U	0	0	0		0	0	0]	0	0	0		0	0	0		0	0	0	L	0	0	0		0	0	0		0	0	0]]

By Definition 4, we can conclude that the inverse of M generated by Algorithm COMPUTE-INV is consistent with that of manual reasoning. Therefore, the proposed algorithm can calculate INV(M) correctly. We compared the result of Algorithm COMPUTE-INV with that of manual reasoning for each basic cardinal direction relation. We find that the results of this algorithm are in good agreement with those of manual reasoning without exception. The results of comparison show that Algorithm COMPUTE-INV is correct and complete.

From the above example, we know that the manual reasoning has to find out all the possible spatial configurations between the primary region and reference region. It is easy to miss some possible case. There needs to make a lot of hypotheses and inferences with the spatial configurations between the primary region and reference region. To make sure there are no omissions, the results need to be checked completely and thoroughly. Obviously, the manual reasoning is a very complicated operation and requires a lot of work. The proposed algorithm COMPUTE-INV can calculate the inverse of any basic cardinal direction relation directly which does not need to use reference tables and any manual operation.

We know that the time cost of this algorithm depends on that of the depth-first search. The search space is a binary tree with depth 9 which only has  $2^9 = 512$  elements. During this search, when a Boolean  $3 \times 3$  matrix is obtained, Algorithm VALID-CDM is performed where at most 9 elements are processed by means of the queue. By Theorem 2, we know that the depth-first search is performed at most 16 times. The time



Figure 7: All possible spatial configurations of b with respect to a.

complexity of algorithm COMPUTE-INV is constant. Therefore, it is unnecessary to concentrate on the time efficiency of Algorithm COMPUTE-INV.

In this section, the correctness and completeness of algorithm COMPUTE-INV has been verified. So far, the problem of computing the inverse of basic cardinal direction relations in the mode of Goyal is solved.

# 8 Conclusion and future work

In this work, we have addressed on the problem of automatically computing the inverse of cardinal direction relations in the model of Goyal. We first provided a solution by theorems for this problem. Then, we gave an algorithm for deciding whether a Boolean matrix is a valid direction matrix or not and an algorithm for computing the original matrices of the rectangular direction matrix. Finally, we provided our algorithm for computing the inverse of cardinal direction relations by means of the operations of direction matrix including the inverse operation of interval matrix, projection of direction matrix, and multiplication of matrix. Theorems were provided to prove formally that our algorithms are correct and complete. The verification was carried out by comparing the result of our algorithm with that of manual reasoning. The results of comparison also demonstrate that our algorithm can work correctly.

Notice that our solution does not need any manual reasoning and calculation, and the help of the reference tables. The proposed algorithm completely realized the automatic inference and calculation of the inverse of the basic cardinal direction relations in the model of Goyal, which improved the ability of intelligent reasoning and prediction of this model and then enhanced the usability of this model. This work is of great significance for improving the ability of spatial reasoning and spatial analysis of spatial database.

However, the real world is a three-dimensional space. In a recent paper [11], we provided a model for cardinal directions in three-dimensional space, which is the extension of the model of Goyal in the threedimensional space. We found that this model has as many as 38,209,336 basic cardinal direction relations by programming calculation. For such a large number of direction relations, it is undoubtedly a huge challenge to perform spatial reasoning with cardinal directions in this model. We plan to extend our algorithms to solve the problem of computing the inverse of cardinal direction relations defined by the above three-dimensional model in a separate paper.

**Acknowledgments:** The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

**Funding information:** This work is partially supported by the science technology department of Henan Province under Grant 222102320349, and the National Natural Science Foundation of China under Grant 61802115.

Conflict of interest: Authors state no conflict of interest.

# References

- [1] Wang M, Wang XT, Li S, Hao ZX. Reasoning with the original relations of the basic 2D rectangular cardinal direction relation. J Xi'an Jiaotong Univ. 2020;54(4):133–43.
- [2] Gong X, Xie Z, Zhou L, He ZJ. The measurement method of spatial orientation similarity binary model. Chin J Surv Mapp. 2021;50(12):1705–16.
- [3] Chen C, Wang ZH, Ma P. Formal description and calculation of the spatial orientation relationship of surface group (Group) targets. Surv Mapp Spat Geogr Inf. 2021;44(9):17–21.
- [4] Christian F, Jasper VDV, Diedrich W. Formal representation of qualitative direction. Int J Geogr Inf Sci. 2018;32(12):2514–34.

- [5] Papadias D, Sellis T. Qualitative Representation of Spatial Knowledge in Two Dimensional Space. VLDB J. 1994;3(4):479-516.
- [6] Skiadopoulos S, Koubarakis M. Composing cardinal directions relations. Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases. CA, USA: Redondo Beach; 2001. p. 299–317.
- [7] Goyal RK, Egenhofer MJ. Similarity of cardinal directions. Proceedings of the 7th International Symposium on Advances in Spatial and Temporal Databases. CA, USA: Redondo Beach; 2001. p. 36–58.
- [8] Liu WM, Zhang XT, Li SJ, Ying MS. Reasoning about cardinal directions between extended objects. Artif Intell. 2010;174(12):951–83.
- [9] Yusuf I, Esra E. Qualitative reasoning about cardinal directions using answer set programming. Proceeding of the Thirty-Second AAAI Conference on Artificial Intelligence. LA, United states: New Orleans; 2018. p. 671–95.
- [10] Skiadopoulos S, Koubarakis M. Composing cardinal direction relations. Artif Intell. 2004;152(2):143-71.
- [11] Wang M, Hao ZX. Qualitative representation and reasoning on direction relation of three-dimension space. Comput Eng. 2009;35(15):22–5.
- [12] Chen J, Jia HY, Liu DY, Zhang CH. Composing cardinal direction relations based on interval algebra. Int J Software Inf. 2010;4(3):291–303.
- [13] Li S, Zhang LP, Hao XH, Hao ZX. Representation and compound of vague region relations and direction relations. J Comput Dev. 2015;52(4):918–28.
- [14] Wang M, Liu XD, Li SY, Li S. Composing 3D cardinal direction relation. J Comput Theor Nanosci. 2016;13(1):623-7.
- [15] Hao XH, Li S, Hao ZX. Representation and reasoning of 3DR46 model in complex 3D space. Comput Sci Explor. 2020;14(12):2004–13.
- [16] Wang J, Jiang GW, Guo R. Research for inverse operation of spatial direction relation. J Geomat Sci Technol. 2008;25(5):324–8.
- [17] Wang M, He L, Li S. Research on inversing the basic cardinal direction relation. Appl Res Comput. 2013;30(1):138-41.
- [18] Wang M, Li L. Reasoning with the inverse of 3D rectangular cardinal direction relations. ICIC Exp Lett Part B Appl. 2013;4(3):581–7.
- [19] Wang M, Huang ZG, Song LI. Reasoning with the inverse of 3D cardinal direction relations based on block algebra. J Comput Appl. 2014;34(4):1144-8.
- [20] Spiros S, Manolis K. On the consistency of cardinal direction constraints. Artif Intell. 2005;163(1):91–135.
- [21] Liu YS, Hao ZX. Consistency checking for cardinal direction relations based on MBR. J Software. 2006;17(5):976–82.
- [22] Navarrete I, Morales A, Sciavicco G. Consistency checking of basic cardinal constraints over connected regions. Proc of 20th International Joint Conference on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann Publishers Inc; 2007. p. 495–500.
- [23] Liu WM, Li SJ. Reasoning about cardinal directions between extended objects: The NP-hardness result. Artif Intell. 2011;175(18):2155–69.
- [24] Goyal R, Egenhofer MJ. The direction relation matrix:a representation for directions relations between extended spatial objects. Bar Harbor Maine: The Annual Assembly and the Summer Retreat of University Consortium for Geographic Information Systems Science; 1997.
- [25] Goyal R, Egenhofer MJ. Cardinal direction between extended spatial objects. IEEE Tran on Know Data Eng. 2000. http:// www.spatial.maine.edu/~max/RJ36.html.
- [26] Wang M, Liu XL, Liu YJ. A model for cardinal direction relations in 3D space. ICIC Express Lett. 2013;7(2):389-96.
- [27] Cicerone S, Di Felice P. Cardinal relations between regions with a broad boundary. Proceeding of the 8th ACM Symposium on Advances in Geographic Information. New York, NY: Association for Computing Machinery; 2000. p. 15–20.
- [28] Dong YQ, Xu WX, Liu JD, Wang SH, Jiang HN. Approach to modeling direction relations between regions with uncertain boundaries. J Beijing Univ Posts Telecommun. 2016;39(1):19–23.
- [29] Lu XM, Yan HW, Wang ZH. The modeling of spatial direction relationship between object groups. J Geo inf Sci. 2018;20(6):721–9.
- [30] Liu YS, Hao ZX. Operation for cardinal direction relations of MBR-based. J Harbin Inst Technol. 2007;39(11):1796-8.
- [31] Wang M, Hao ZX. Location representation and cardinal direction relation reasoning based on qualitative coordinates. J Xi'an Jiaotong Univ. 2010;44(8):36–42.
- [32] Tang XH, Qin K, Meng LK. A qualitative matrix model of direction-relation based on topological reference. Acta Geodaetica Cartograph Sin. 2014;43(4):396–403.
- [33] Li PP, Liu JP, Yan HW, Lu XM. An improved model for calculating the similarity of spatial direction based on direction relation matrix. J Geomat Sci Technol. 2018;35(2):216–20.
- [34] Allen JF. Maintaining knowledge about temporal intevrals. Commun ACM. 1983;26(11):832-4.
- [35] Balbiani P, Condotta J, Fari~nas del Cerro L. A model for reasoning about bidimensional temporal relations. Proceedings of the 6th International Conference on Principles of Knowledge Representation and Reasoning. Trento, Iatly; 1998.
   p. 124–30.
- [36] Balbiani P, Condotta J, Fari nas del Cerro L. A new tractable subclass of the rectangle algebra. Proceedings of the 16th International Joint Conference on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann Publishers Inc. 1999.
   p. 442–7.