

Cardinal Directions between Complex Regions

MARKUS SCHNEIDER, TAO CHEN, GANESH VISWANATHAN, WENJIE YUAN
University of Florida

Besides topological relationships and approximate relationships, *cardinal directions* like *north* and *southwest* have turned out to be an important class of qualitative spatial relationships. They are of interdisciplinary interest in fields like cognitive science, robotics, artificial intelligence, and qualitative spatial reasoning. In spatial databases and Geographic Information Systems (GIS) they are frequently used as join and selection criteria in spatial queries. However, the available computational models of cardinal directions suffer from a number of problems like the use of too coarse approximations of the two spatial operand objects in terms of single representative points or minimum bounding rectangles, the lacking property of converseness of the cardinal directions computed, and the limited applicability to *simple* instead of *complex* regions only. This article proposes and formally defines a novel two-phase model, called the *Objects Interaction Matrix (OIM)* model, that solves these problems, and determines cardinal directions for even complex regions. The model consists of a tiling phase and an interpretation phase. In the *tiling phase*, a tiling strategy first determines the zones belonging to the nine cardinal directions of each individual region object and then intersects them. The result leads to a bounded grid called *objects interaction grid*. For each grid cell the information about the region objects that intersect it is stored in an *objects interaction matrix*. In the subsequent *interpretation phase*, a well defined interpretation method is applied to such a matrix and determines the cardinal direction. Spatial example queries illustrate our new cardinal direction concept that is embedded in a spatial extension of SQL and provides user-defined cardinal direction predicates.

Categories and Subject Descriptors: H.2.8 [Database Management]: Spatial databases and GIS; H.2.3 [Database Management]: Query languages

General Terms: Design, Languages, Management

Additional Key Words and Phrases: Cardinal direction, directional relationship, spatial database

1. INTRODUCTION

In recent years, the exploration of *cardinal directions* between objects in space has turned out to be a multidisciplinary research issue involving disciplines like artificial intelligence, cognitive science, geographic information science/systems (GIS), linguistics, robotics, spatial analysis, spatial database systems, and qualitative spatial reasoning. Cardinal directions are an important spatial concept and describe purely qualitative properties that characterize the *directional relationships* of spatial objects to each other. More precisely, they represent *absolute* directional relationships

Authors' address: M. Schneider, T. Chen, G. Viswanathan, W. Yuan, University of Florida, Department of Computer & Information Science & Engineering, PO Box 116120, Gainesville, FL 32611-6120, USA.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2011 ACM 0362-5915/2011/0300-0001 \$5.00

like *north* and *southwest* with respect to a given reference or coordinate system in contrast to *relative* directional relationships like *in_front* and *left*. From a database and GIS perspective, research efforts have been motivated by the necessity of computational and formally defined models of *directional predicates*. These predicates are employed as filter conditions for spatial selections and spatial joins in spatial query languages and as a support for spatial data retrieval and analysis tasks, both at the user level for reasons of conceptual clarity and at the query processing level for reasons of efficiency. They enable the user, for example, to ask for all hurricanes that have ever been west of Florida, the general direction of the whale routes in the Gulf of St. Lawrence, or all land parcels that are located to the west of the parcels of power stations. The background for the interest in latter query, for example, could be that due to west winds these parcels are endangered by the hazardous influence of air pollution. Assuming a relation *land_parcel* with attributes *id* of type *integer* and *parcel* of type *region* as well as a relation *power_station* with attributes *name* of type *string* and *loc* of type *region*, we can express the query example, which represents a *directional join* as a special case of a spatial join, in an SQL-like style as follows:

```
SELECT id, name FROM land_parcel, power_station WHERE parcel west_of loc
```

In the literature, several approaches have been proposed to model cardinal directions. The more expressive and more precise approaches have been exclusively designed for region objects. As we will show in this article, unfortunately all of them suffer from at least one of three main problems. First, some models do not consider the shapes of the two operand objects of a cardinal direction relation but use quite coarse approximations and simplifications in terms of single representative points or minimum bounding boxes (*object shapes consideration requirement*). This leads to inaccurate results. Second, some models yield inconsistent results for the cardinal direction $dir(A, B)$ between two spatial objects A and B and its converse cardinal direction $dir(B, A)$ in the sense that one result is not always the inverse of the other (*converseness requirement*). That is, it should hold that $dir(B, A) = inv(dir(A, B))$ and $dir(A, B) = inv(dir(B, A))$. For example, if a cardinal direction model ascertains that A is located to the west of B , then the model should also correctly ascertain that B is located to the east of A , and vice versa. For our query example, this means that the query

```
SELECT id, name FROM land_parcel, power_station WHERE loc east_of parcel
```

must always yield the same result as the query above. Third, some models only work well if the region objects involved in direction computations have a *simple* structure. This means that they only consist of one component, are connected, and do not have holes. However, this is in contrast to the common consensus in the spatial database community that simple geometric structures are inadequate abstractions for real spatial applications since they are insufficient to cope with the variety and complexity of geographic reality. *Complex* region objects are needed that allow multiple components and may have holes. As a consequence, some models can yield wrong or counterintuitive results for certain spatial scenarios that involve complex region objects (*support of complex objects requirement*).

The goal of this paper is to propose and design a computational model of cardinal directions and directional predicates for complex regions that overcomes the limitations of available approaches. This requires that the model takes better into account the shape of the region operand objects, ensures the property of converseness of cardinal directions (that is, for a directional relationship p holds that $A \ p \ B \Leftrightarrow B \ inv(p) \ A$), accepts complexly structured region objects as operands, and avoids the wrong results computed by some approaches.

Our solution consists in a novel two-phase method that includes a *tiling phase* followed by an *interpretation phase*. In the first phase, we apply a tiling strategy which first determines those zones that correspond to nine cardinal directions with respect to *each* region object. The zones of both objects are then intersected. The result leads to a closed grid that we call *objects interaction grid* (*OIG*). For each grid cell we derive the information about the region objects that intersect it and store this information in a so-called *objects interaction matrix* (*OIM*). In the second phase, we apply an *interpretation method* to such a matrix and determine the cardinal direction. Query examples illustrate our concepts.

The remainder of this article is organized as follows: Section 2 discusses related work. It sketches the spatial data type *region* as the operand type of cardinal directions and directional relationships, and in detail classifies, describes, and compares the available cardinal direction models with respect to specified design criteria. Section 3 provides a motivation and coarse overview of our own, two-phase method, called *Objects Interaction Matrix* (*OIM*) model, to this computational problem. Its tiling phase is described in Section 4, and its interpretation phase is delineated in Section 5. Section 6 analyzes the OIM model and compares it to past approaches. Section 7 derives directional predicates from cardinal directions and shows how these predicates can be used in spatial database queries. Finally, Section 8 draws some conclusions and depicts future work.

2. RELATED WORK AND A COMPARATIVE STUDY OF CARDINAL DIRECTION MODELS

In this section, we discuss the literature of cardinal direction models. Section 2.1 provides a brief overview of the spatial data type *region* which represents complex region objects that are the operands of the cardinal directions and directional predicates considered in this article. In Section 2.2, we pose essential design requirements for cardinal directions that are partially taken from the literature and partially derived from the weaknesses found in the existing cardinal direction models. In Section 2.3, we analyze the available cardinal direction models. Section 2.4 describes the few available designs of directional predicates in databases. Finally, in Section 2.5, we give a summary of the limitations of the related work as well as an overview of our novel contributions.

2.1 The Spatial Data Type *region*

In the spatial database and GIS communities, *spatial data types* like *point*, *line*, or *region* (Figure 1) have found wide acceptance as fundamental abstractions for modeling the structure of geometric entities, their relationships, properties, and operations. In this article, we confine ourselves to region objects since only for them (and for point objects) cardinal direction models have been proposed. The

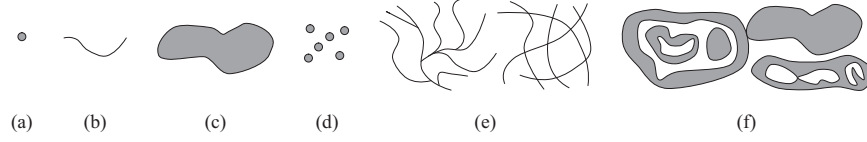


Fig. 1. Examples of a simple point object (a), a simple line object (b), a simple region object (c), a complex point object (d), a complex line object (e), and a complex region object (f).

literature distinguishes *simple* region objects (*simple* spatial data types in general [Egenhofer 1994; Güting 1988]) and *complex* region objects (*complex* spatial data types in general [Clementini and Di Felice 1996; Schneider 1997; Schneider and Behr 2006; Worboys and Bofakos 1993]), depending on the spatial complexity they are able to model. Simple regions (Figure 1(c)) represent areal, bounded, and connected spatial objects. Complex regions (Figure 1(f)) allow multiple components, permit holes, and are closed under the geometric set operations *intersection*, *union*, and *difference* [Schneider 1997]. The capability of dealing with complex region objects is an important requirement for any suitable cardinal direction model.

2.2 Design Requirements for Cardinal Direction Models

We now identify and formally define design criteria for cardinal direction models and check these criteria in Sections 2.3 and 2.5 against the available models and our new Objects Interaction Matrix model. The degree of fulfilment of each design criterion is evaluated for each model by up to three different assessment values named as *full support*, *partial support*, and *no support*. The literature [Abdelmoty and Williams 1994; Frank 1996] postulates four general design criteria for a model of binary qualitative spatial relationships: *completeness*, *soundness*, *uniqueness*, and *generality*. In detail, they are specified as follows:

Completeness. A model must be able to determine the spatial (in our case: directional) relationship for any configuration of two spatial objects, that is, the model must be exhaustive. Formally, let M be a cardinal direction model, and let $dir_M(A, B)$ denote the function for M that yields the cardinal direction of two spatial objects A and B . Further, let CD_M denote the set of cardinal directions that M is supposed to distinguish. We say that M provides *full support* for the completeness requirement if the statement holds that $\forall A, B \in region : dir_M(A, B) \in CD_M$. If this is not the case, then M is said to provide *no support* for the completeness requirement.

Soundness. Only correct and realizable qualitative relationships should be derived from the model. For directional relationships this especially means that the result comprises all existing basic cardinal directions. Let dir_M also be applicable to any two points $p, q \in \mathbb{R}^2$ and thus provide the precise and correct cardinal direction between those points. We say that a model M provides *full support* for the soundness requirement if $dir_M(A, B) = \bigcup_{p \in A, q \in B} dir_M(p, q)$. M provides *no support* for the soundness requirement if $dir_M(A, B) \cap \bigcup_{p \in A, q \in B} dir_M(p, q) = \emptyset$. Finally, M is said to provide *partial support* for the soundness requirement if $dir_M(A, B) \cap \bigcup_{p \in A, q \in B} dir_M(p, q) \neq \emptyset \wedge dir_M(A, B) \neq \bigcup_{p \in A, q \in B} dir_M(p, q)$.

Uniqueness. All qualitative relationships between spatial objects should be uniquely

distinguishable by the model, that is, the possible relationships should be mutually exclusive. For example, if the cardinal direction between two spatial objects is *north* according to the model, then *north* must be the only true cardinal direction between these objects. All other cardinal directions must then be false. If this is the case, we say that a model offers *full support* for the uniqueness requirement; otherwise, it offers *no support*.

Generality. A model must be able to determine the cardinal directions for spatial objects of different shapes, that is, it must be applicable to all combinations of the spatial data types *point*, *line*, and *region*. If this is the case, we say that a model offers *full support* for the generality requirement; otherwise, it only offers *partial support*.

In addition to the above requirements, we have identified three further design criteria that are specific to cardinal directions and that result from the weaknesses of the existing models described in Section 2.3. They include the *object shapes consideration*, *converseness*, and the *support of complex objects*. In detail, they can be characterized as follows:

Object shapes consideration. A model should be able to leverage the shape of the interacting operand objects to increase the quality and precision of the deduced cardinal directions. A model offers *full support* for the object shapes consideration requirement if the cardinal direction is computed with respect to the exact shapes of both operand objects. However, no existing model has this property since they are all based on approximations of spatial objects. Different approximation levels yield results with different precisions. We say a model provides *no support* for the object shapes consideration requirement if the operand objects are approximated with lower-dimensional objects (for example, centroids) that ignore the extent of an operand object. A model provides *partial support* for this requirement if the operand objects are represented by approximation objects of the same dimension (for example, minimum bounding rectangles or circles).

Converseness. A model must ensure the converseness of cardinal directions. For example, if a model M determines that a spatial object A is southeast of a spatial object B , then M must also determine that B is northwest of A . Formally, this means that $dir_M(B, A) = inv(dir_M(A, B))$ must hold where $inv(D) = \{inv(d) \mid d \in D\}$ for $D \subseteq CD_M$. Further, $dir_M(A, B) = inv(inv(dir_M(A, B)))$ must hold. If M fulfills these properties, it is said to offer *full support* for the converseness requirement. Otherwise, M offers *no support*.

Support of complex objects. A model should not only be able to determine the cardinal direction of simple spatial objects but especially of complex spatial objects (Section 2.1) since only the latter represent spatial reality and are included in spatial data representation standards nowadays. If this is the case, the exact set of cardinal directions between complex objects includes all possible cardinal directions between each component pair of the interacting objects. We then say that the model offers *full support* for the support of complex objects requirement. If a model can implicitly support complex objects by simply applying it to them but does not provide all valid cardinal directions, we say that it offers *partial support* for this design criterion.

2.3 Evaluation of Existing Cardinal Direction Models

Several models have been proposed to capture cardinal direction relations between spatial objects as instances of spatial data types. These models can be broadly classified into *tiling-based* models (Section 2.3.1), *interval-based* models (Section 2.3.2), and *vector algebra-based* models (Section 2.3.3).

2.3.1 Tiling-based Cardinal Direction Models. *Tiling-based* models define cardinal directions by using so-called *partitioning lines* that subdivide the plane into semi-open tiles. The models can be further subdivided into *projection-based* models and *cone-shaped* models. *Projection-based* models create partitioning lines parallel to the coordinate axes while *cone-shaped* models partition the space into angular zones. Depending on how objects are used to create the partitions, each tiling-based model can be classified as either asymmetric tiling that uses only one object for tiling, or as symmetric tiling which uses both objects for tiling.

Projection-based models define cardinal directions by using partitioning lines parallel to the coordinate axes. The early approaches degenerate one or both extended spatial operand objects (that is, lines with their linear extent and regions with their areal extent) to single representative points. One representative point plays the role of a *reference object*, and the other representative point plays the role of a *target object*. The tiling is here only performed with respect to the reference object; hence, we call it *asymmetric tiling*. The cardinal direction is then determined from the target object towards the reference object. Frank’s approach [Frank 1991; 1992; 1996] approximates spatial objects by their center points and defines cardinal directions by using partitioning lines parallel to the coordinate axes. The space is divided into four non-overlapping zones by the two partitioning lines that cross the reference point. This model uses the composition of two adjacent basic cardinal directions, that is, *north*, *south*, *west*, and *east*, in each zone to assign one of the four pairwise opposite directions *northwest*, *northeast*, *southeast*, and *southwest* to it. The direction is also determined by the zone in which the target point falls. The point approximation suffers from the problem that the generalization step completely ignores the shape and the extent of the spatial operand objects and thus leads to easy to use but rather inaccurate models since the operand objects can intersect several zones while their reference points fall into a single and unique zone. Thus the *object shapes consideration* requirement is not supported.

The *Direction-Relation Matrix (DRM)* model [Goyal and Egenhofer 2000a; 1997; Skiadopoulos and Koubarakis 2004] presents a major improvement of this approach. In this model, the reference object is responsible for creating the partition zones. The partitioning lines are given by the infinite extensions of the segments of its minimum bounding rectangle. This leads to a tiling with the nine zones of *north*, *west*, *east*, *south*, *northwest*, *northeast*, *southwest*, *southeast* as well as a *central zone* named *sameLocation* and is given by the minimum bounding rectangle of the reference object. The target object contributes with its exact shape, and a Boolean *direction-relation matrix* stores for each tile whether it is intersected by the target object (see Figure 2a). This model also employs the *asymmetric tiling* strategy. In the DRM model, spatial operand objects are not simplified to point objects. Further, the minimum bounding rectangle of the reference object is used with the shape of the target object to determine the cardinal direction. Thus, this

model better supports the *object shapes consideration* requirement. However, this model can lead to counterintuitive computations of cardinal directions and prevent us from consistently determining the converse cardinal direction. This violates the *converseness* requirement. We give an example of these weaknesses of the DRM model in our case study in Section 6. An extension of this model, called the *Deep Direction-Relation Matrix (DDRM)* model [Goyal and Egenhofer 2000b], is able to also handle point and line features but has inherited the weaknesses of the DRM model.

Our *Objects Interaction Matrix (OIM)* model also belongs to the projection-based models. Its strategy and properties as well as its comparison to other models will be given in detail in later sections. In [Chen et al. 2010] we have already given a first introduction of this model. We describe the fundamental concepts of the two-phase OIM model with its tiling phase and interpretation phase and sketch an SQL query language extension with embedded directional predicates. In this article, we elaborate and formalize the OIM concept, study its expressiveness by determining the valid spatial scenarios of two complex region objects for which cardinal directions can be ascertained, formally define the semantics of the directional result that the OIM model computes, systematically derive directional predicates from the OIM model, design SQL data definition language constructs that enable the user to specify user-defined and composite directional predicates, and demonstrate the embedding of these predicates in SQL queries.

Cone-shaped models define directional relations by using angular zones. The early approaches also generalize a reference object and a target object by point objects. The first cone-shaped point approximation model attempting to derive the cardinal direction between two spatial objects in a qualitative way is Haar’s *cone-shaped point model* [Haar 1976]. It defines cardinal directions by using partitioning lines with an angle of 45° through a reference point, which is the center point of the reference object. The space is divided into four mutually exclusive zones called *north*, *south*, *west*, and *east*. The cardinal direction between the two spatial operand objects is determined by the zone in which the target point (the center of the target object) falls. Since the subdivision into only four zones may lead to coarse results, the approach in [Jing et al. 2005] provides a slight extension of the model in [Haar 1976]. It subdivides each of the four zones of the cone-shaped model into three smaller zones. Hence, we obtain twelve, more fine-grained directions, and the cardinal direction between two spatial objects can be more accurately determined.

Peuquet & Zhang [Peuquet and Zhang 1987] point out that the choice of representative points (like center points) can have a negative effect on the correctness of the cardinal direction determined for two extended spatial objects since an extended spatial target object can fall into several different zones while a representative point always falls into one zone. They propose a method to choose the representative point of the object by better considering the shapes of the objects. They first test if the minimum bounding rectangle of the target object falls within a 90° triangular area extended from the centroid of the reference object (See Figure 2b). Depending on this, the vertex of the triangle is moved backward or forward to find the most appropriate representative point for the reference object. The 90° triangles extending from the new representative point are used to determine the relative direction

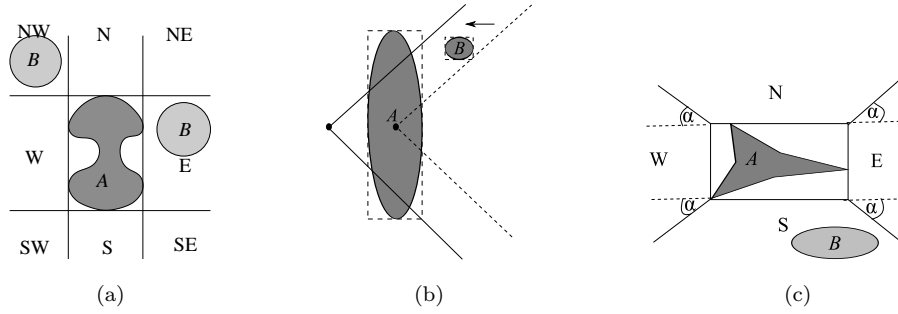


Fig. 2. The Direction-Relation Matrix model tiling with A as reference object (a), Peuquet's model (b), and the Cone-shaped Directional Relations model family with variable angles α (c)

between the two objects.

All approaches that generalize objects into points share the same problems and violate the same requirements that we have discussed for the projection-based models. The *Cone-shaped Directional Relations* (CDR) model family [Skiadopoulos et al. 2007] is an improvement of the model in [Haar 1976] and uses the minimum bounding rectangle of the reference object to subdivide the space around it with partitioning lines emanating from the corners of the rectangle with different angles. That is, this model employs the *asymmetric tiling* strategy. By adjusting the angles, an infinite number of models can be represented for different applications (Figure 2c). The partitioning lines subdivide the space into five directional zones called *south*, *east*, *west*, *north*, and *center area*. The cardinal direction is determined by the zone in which the target object falls.

Abdelmoty & Williams [Abdelmoty and Williams 1994] present an intersection-based extension to the cone-shaped approach in [Peuquet and Zhang 1987]. Their extension applies a *symmetric tiling* strategy, approximates an object by its minimum bounding rectangle, and represents it in terms of its components, namely, the interior, the exterior, and the boundary, by employing the triangular model. Arbitrary objects are generally approximated by bounding boxes but point abstraction is also allowed. Angular extensions emanating from the corners of the minimum bounding rectangle of an object enable the creation of four semi-infinite areas around each of them. The boundary of each area is represented by two direction lines (for example, NW and NE for the northern direction) and a characteristic point or a side of the minimum bounding rectangle (for example, the top of the minimum bounding rectangle for the northern direction). The directional relationship between two objects is then given as a result of the combinatorial intersection of the components and evaluated using an intersection matrix. For example, A is to the west of B if the boundary of the area representing the east of A has an overlap with the boundary of the region representing the west of B . Thus, the intersection-based methodology is used for the representation of cardinal directions between objects of arbitrary shapes and sizes. However, no consideration is given to intertwined objects with overlapping bounding boxes, thus the *completeness* requirement is violated. Moreover, the model can yield incorrect results when the extent of an object is disregarded due to the use of point abstraction.

The model by Safar & Shahabi [Safar and Shahabi 1999] describes another tiling approach to computing the cardinal directions between spatial objects. It uses *minimum bounding circles (MBC)* as object approximations for both region objects. This approach pursues an asymmetric tiling strategy and hence distinguishes between a reference object and a target object. The plane is partitioned into five zones by overlaying the cone-shaped directions of the reference object [Haar 1976; Peuquet and Zhang 1987] with its minimum bounding circle called *same level*. The cardinal direction of any target object with respect to the reference object is then given by the zones in which the minimum bounding circle of the target object lies. But due to the asymmetric tiling, this method does not provide converse directions.

2.3.2 Interval-based Cardinal Direction Models. *Interval-based* models make use of the x -intervals and y -intervals of the minimum bounding rectangles of both spatial operand objects, pairwise apply Allen's thirteen interval relations [Allen 1983] to them, and then derive the cardinal direction from the result.

The *2D-string* models [Chang and Li 1988; Chang 1989] use a pair of relationship strings as a symbolic representation of the projections of two region objects in a picture (image) onto the two coordinate axes. The cardinal directions are generated relative to a predefined grid superimposed on the picture. An extension of this approach is based on *2D projection interval relationships (PIR)* [Nabil et al. 1995]. It uses a graph representation instead of a string for picture matching and spatial reasoning purposes and can determine both directional and topological relationships between all possible spatial object pairs in a picture using Allen's interval relationships. The *2D+ string* extension [Kim and Um 1999] uses metadata to describe spatial relationships between objects in a picture. The model basically uses six basic direction elements defined by using the extensions of the minimum bounding rectangle of an object A . A 2D+ string is formed by a pair of one-dimensional strings, one string for each of the two axes. Both one-dimensional strings together permit the inference of directional and topological relationships between spatial objects. The 2D-PIR and 2D+ string models cannot handle all spatial scenarios, and hence violate the *completeness* requirement. In the 2D-string model, the size of equal-sized grid becomes very important; if it is too large, the model loses precision, and if it is too small, an object gets subdivided into more cells than needed.

Papadias *et al.* propose another model [Papadias and Egenhofer 1997; Papadias and Sellis 1994; Papadias and Theodoridis 1997] that also makes use of the minimum bounding rectangles of both operand objects and that also applies Allen's thirteen interval relations to the rectangle projections on the x -axis and the y -axis respectively. A complete set of $13 \times 13 = 169$ possible configurations of two rectangles is obtained [Papadias et al. 1996], which is expressive enough to cover all cardinal directions between two minimum bounding rectangles. But this model can yield misleading and ambiguous cardinal directions if spatial objects are overlapping, intertwined, or horseshoe shaped. Thus, it does not support the *object shapes consideration* requirement very well. Papadias's model has an overlapping set of basic and higher level directional relations. Hence a directional relationship provided does not imply the negation of the other directional relationships. For example, the *strong_north* and *north* relationships are not mutually exclusive. Thus, it violates the *uniqueness* requirement.

2.3.3 Vector Algebra-based Cardinal Direction Models. The authors in [Shekhar et al. 1999] provide a conceptual model that is fundamentally different from all the other models described so far in the sense that it models direction as a spatial object and not as a relational predicate between spatial objects. This model uses concepts such as vectors, points, and angles. The basic approach is to model direction as a unit vector, which is a spatial object, and orientation as a set of directions. This leads to a simple vector algebra that can also be used to perform direction reasoning. However, this approach is restricted to point objects. Extended objects are approximated by their centroids. Thus, the *object shapes consideration* requirement is not supported, which leads to an inaccuracy of the model.

2.3.4 Ternary Projective Relations. All publications mentioned so far and also the Objects Interaction Matrix model deal with *absolute* directional relationships like *north* and *southwest*. They view spatial objects from a *global* perspective in a given, fixed reference or coordinate system. Another important class of directional relationships are *relative* directional relationships like *in-front-of*, *above*, and *left-of*. They view spatial objects from a *local* perspective in a variable reference or coordinate system. We do not cover them in this article since they require rather different models and refer usually in applications to spatial objects in the three-dimensional space. They require an *observer* who looks at a spatial scenario of spatial objects and assesses the relative directional relationships between them by projections. These relationships are therefore also called *ternary projective relations* [Clementini and Billen 2006; Isli and Cohn 2000].

2.4 Directional Predicates for Querying in Spatial Databases

A cardinal direction model provides a method for the *determination* of the cardinal direction between two spatial objects. However, in a database context, cardinal directions are not directly applicable. Instead, in queries, *directional relationships* are needed as Boolean predicates for the *validation* of cardinal directions between two spatial objects. These predicates can be employed as filter conditions for spatial selections and spatial joins in query languages, and as a support for spatial data retrieval and analysis operations.

The model by Papadias (Section 2.3.2) has made some proposals for the definition and integration of directional predicates into database queries. This model defines several groups of directional relationships for the user at two different levels of precision. At a higher level of precision, nine mutually exclusive, complete, and basic directional relationships are defined for *any two points* of two spatial objects, namely, *north-west*, *restricted_north*, *north-east*, *restricted_west*, *same_position*, *restricted_east*, *south-west*, *restricted_south* and *south-east*. At a lower level of precision, additional directional relationships are then specified as disjunctions of the basic nine relationships such as *north* (including *north-west*, *restricted_north*, and *north-east*) and correspondingly *east*, *south*, and *west*. Further, two special directional relationships named *same_level* (including *restricted_west*, *same_position*, and *restricted_east*) and *same_width* (including *restricted_north*, *same_position*, and *restricted_south*) are defined. Using these definitions, a set of directional relationships between two spatial objects is defined that considers different nuances of a particular cardinal direction. For exam-

ple, for the northern cardinal direction, the model provides the directional relationships *strong_north*, *weak_north*, *strong_bounded_north*, *weak_bounded_north*, *strong_north_west*, *weak_north_west*, *strong_north_east*, *weak_north_east*, *north*, and *just_north*. Due to the minimum bounding rectangle approximation, this approach can give coarse cardinal directions between spatial objects.

The authors in [Zhu et al. 2001] use a reference object/target object model similar to the Direction-Relation Matrix Model (see Figure 2a) and propose three different categories of directional relationships. They distinguish (i) *uni-direction predicates* like *NW* and *S* if the target object is located in exactly one of the nine zones (tiles), (ii) *bi-direction predicates* like *NE-N*, *NE-E*, *SW-S*, and *SW-W* if the target object intersects two horizontally or vertically adjacent tiles, and (iii) *tri-direction predicates* like *weak-W* and *weak-E* (corresponding to *western* and *eastern*) if the target object intersects three horizontally or vertically adjacent tiles. Our OIM model includes all these predicates.

In Section 7, we will show how directional predicates can be systematically defined based on our OIM model and how they can be integrated into the database query language SQL.

2.5 Summary of Related Work Limitations and Novel Contributions

Based on the general design criteria of *completeness*, *soundness*, *uniqueness*, and *generality* identified in the literature for models of binary qualitative spatial relationships, the three further design requirements of *objects shape consideration*, *converseness*, and *support of complex objects* in Section 2.2, and our discussion of the available qualitative cardinal direction models and their drawbacks in Section 2.3, Table I provides a summary of the available approaches and checks and compares these models including our Objects Interaction Matrix model against these seven design criteria. The table reveals that the available cardinal direction models fulfil the design criteria of completeness, soundness, and uniqueness quite well while they exhibit shortcomings with respect to the remaining criteria. The Objects Interaction Matrix model, which we propose in this article, promises to fulfil most of the design criteria with a good level of support for each of them. A case study that further illustrates the weaknesses of available cardinal direction models and shows how our OIM model overcomes them is given in Section 6.

Our novel contributions presented in this article can be summarized as follows:

- (1) *Formal definition of a new two-phase cardinal direction model* that satisfies important and well accepted design criteria and separates between (i) a unique and closed tiling strategy to capture the interaction of spatial objects and (ii) the interpretation of the resulting objects interaction matrices to derive the composite cardinal direction.
- (2) *Consistent results when computing the cardinal direction between two objects A and B and between B and A* due to the fulfillment of the converseness property.
- (3) *A higher level of precision in the resultant set of cardinal direction relations compared to any other existing model* due to an improved tiling strategy that intersects the minimum bounding rectangles of *both* input region objects and determines the interaction of each object with respect to the resulting grid.
- (4) *Consideration of complex region objects* as they appear in spatial real world

Models ¹	Completeness	Soundness	Uniqueness	Generality	Shape Consideration	Converseness	Complex Objects
Harr's model	✓	±	✓	±	—	✓	±
Peuquet's model	✓	±	✓	±	—	✓	±
Frank's model	✓	±	✓	±	—	✓	±
2D-string model	—	±	✓	±	—	—	±
Papadias's model	✓	±	—	±	±	—	±
CDR model	✓	±	✓	±	±	—	±
Abdelmoty's model	✓	±	✓	±	—	—	±
MBC model	✓	±	✓	±	±	—	±
DRM model	✓	±	✓	± ²	±	—	±
Shekhar's model	✓	±	✓	±	—	✓	±
OIM model	✓	±	✓	±	± ³	✓	✓

¹ '✓' means the model offers *full support* for the requirement, '±' indicates *partial support*, and '—' shows *no support* (see Section 2.2 for a definition of these terms for each design criterion).

² The DRM model has been extended with support for generality and called the DDRM model (see Section 2.3.1).

³ Our OIM approach uses the shapes of both operand objects and their intersection with the grid tiles to create the objects interaction matrix. Thus it provides a higher level of support for shape consideration when compared to all other models.

Table I. Comparison of the available cardinal direction models including the OIM model against the seven design criteria

scenarios and applications, which is in contrast to available models that are only able to deal with *simple* region objects.

- (5) *An exhaustive evaluation of the valid and thus spatially realizable objects interaction matrices for two complex region objects and two simple region objects respectively* through a three-stage proof technique called *proof-by-constraint-grouping-drawing*.
- (6) *Precise definition and description of cardinal direction predicates* classified into existential predicates and six categories of user-defined predicates
- (7) *Specification of user-defined cardinal direction predicates in SQL* through a user-friendly extension of the SQL Data Definition Language (DDL) as well as a *high-level embedding of these predicates into the SQL query language* supporting direction querying, enabled due to the integration of spatial data types (like *region*) as abstract data types in a database schema, and illustrated by several queries based on a real-world application.
- (8) *Extensive analysis of the shortcomings of the available cardinal direction models* as well as a *detailed comparison of the available cardinal direction models to our Objects Interaction Matrix model in a case study*.

3. OVERVIEW OF THE OBJECTS INTERACTION MATRIX (OIM) MODEL

In this section, we give a brief overview of our novel cardinal direction model, called the *Objects Interaction Matrix (OIM)* model. We emphasize its main features and sketch how it overcomes the weaknesses of current models and how it satisfies the requirements established in Section 2.2. The OIM model belongs to the tiling-based models and here especially to the projection-based models. However, it is important

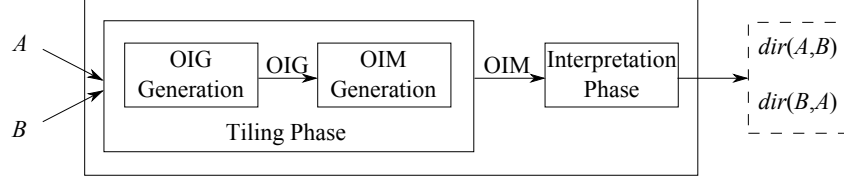


Fig. 3. Overview of the two phases of the Objects Interaction Matrix (OIM) model

to understand that it rather differs from the Direction-Relation Matrix (DRM) model and is thus *not* its extension. For example, the DRM model subdivides the entire unbounded Euclidean space around a reference object, while the OIM model subdivides a closed subspace enclosing both operand objects. Figure 3 shows its two-phase strategy for calculating the cardinal direction between two spatial objects A and B . In the following, we assume that A and B are non-empty values of the complex spatial data type *region* [Schneider 1997].

In the first phase, called the *tiling phase*, we first determine the nine directional zones that belong to region object A (an example is given in Figure 2a) and then the nine directional zones that belong to region object B . We obtain two partitions of the Euclidean plane, and the simple but fundamental, new idea is now that both partitions are overlaid so that the region objects A and B interact with each other in the tiling process. The partition overlay generates a grid called *objects interaction grid* (*OIG*) (Figure 4a). In contrast to all other tiling-based models which have unbounded zones except for the central zone, our grid is closed and bounded. We achieve this by omitting all peripheral, unbounded zones. Regarding our example, a 3×3 -grid is generated (see the continuous segments in Figure 4a). The surrounding 16 unbounded grid cells (indicated by the dashed segments in Figure 4a) are irrelevant since neither A nor B can intersect them. More precisely, the area covered by the objects interaction grid, called *objects interaction grid space* (*OIG space*), is given by the minimum and maximum x - and y -coordinates of the minimum bounding rectangles of A and B . We will later see that also all the other $n \times m$ -grids are possible with $1 \leq n, m \leq 3$.

The partition overlay caters for an equal and symmetric treatment of the operand objects A and B . Concepts like reference object and target object do *not* exist in our model. An objects interaction grid provides us with the valuable information which object intersects which grid cell. This means that a grid cell or tile may be intersected by no region (coded by 0), by region A only (coded by 1), by region B only (coded by 2), or by both regions A and B (coded by 3). For each grid cell $t_{i,j}$ in the i th row and j th column ($1 \leq i, j \leq 3$) we store the coded information about the regions that intersect it in an *objects interaction matrix* (*OIM*) M in cell $M_{i,j}$. That is, we abstract from the geometry of the OIG space and only keep the information which region intersects which tile. We can do this since cardinal directions have a qualitative and not a quantitative nature. In our example, we obtain a 3×3 -objects interaction matrix (Figure 4b).

In the second phase, called the *interpretation phase*, we leverage the objects interaction matrix to derive the cardinal direction between A and B . We call this step *interpretation* since such a matrix can be mapped to different *models of basic*

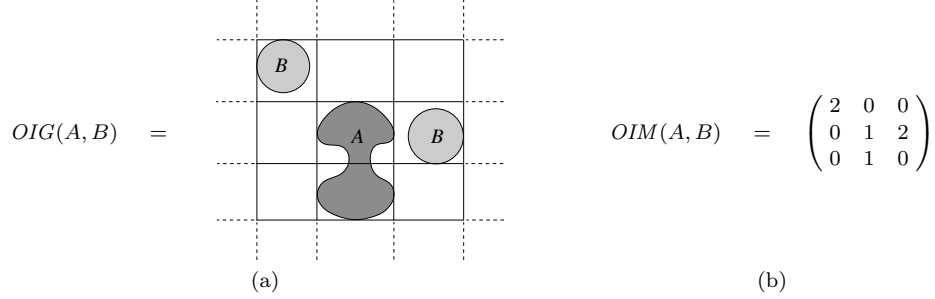


Fig. 4. The objects interaction grid $OIG(A, B)$ for the two complex region objects A and B in Figure 2a (a) and the derived objects interaction matrix $OIM(A, B)$ (b)

cardinal directions, that is, it can be interpreted in different ways. In this article, we have already seen three different models and thus three different interpretations of basic cardinal directions, namely the four cardinal directions *north*, *east*, *south*, and *west*) [Haar 1976], the four cardinal directions *northwest*, *northeast*, *southwest*, *southeast* [Frank 1996], and the nine cardinal directions *north*, *east*, *south*, *west*, *northwest*, *northeast*, *southwest*, *southeast*, and *origin* [Goyal and Egenhofer 1997; Skiadopoulos and Koubarakis 2004]. We will in this article confine ourselves to the latter cardinal direction model due to its popularity and large detailedness, and in order to make our approach comparable to the Direction-Relation Matrix model.

We distinguish two steps in the interpretation phase. In the first step, we use an index pair (i, j) with $1 \leq i \leq m$ and $1 \leq j \leq n$ to represent the location of the element $M_{i,j}$ in the $m \times n$ objects interaction matrix M . We apply a function *loc* to each region and M in order to determine at which locations we can find components of each region. For our example in Figure 4 we obtain $loc(A, M) = \{(2, 2), (3, 2)\}$ and $loc(B, M) = \{(1, 1), (2, 3)\}$. In the second step, we use a function *dir* to determine the *composite cardinal direction* between A and B . We form all pairs of elements of $loc(A, M)$ and $loc(B, M)$ and determine the basic cardinal direction for each pair by applying a corresponding *interpretation function*. The composite cardinal direction between A and B is then equal to the union of all determined basic cardinal directions. The interpretation function determines the basic cardinal direction between any two object components on the basis of their (i, j) -locations in the objects interaction matrix. The values of the interpretation function are stored in an *interpretation table* for a lookup in constant time. For our example in Figure 4, let us consider $(3, 2) \in loc(A, M)$ and $(2, 3) \in loc(B, M)$. The fact that a component of A is located in the cell $(3, 2)$ and a component of B is located in the cell $(2, 3)$ implies that the component of A must be southwest of the component of B . Overall, we obtain $dir(A, B) = \{SW, W, SE\}$ and, similarly, $dir(B, A) = \{NE, E, NW\}$.

Reviewing the goals formulated in the Introduction and in Section 2.2, we see that the shapes of *both* region objects are taken into account. Further, both region objects are treated as equal partners. To a large extent, this coequal treatment contributes to assuring the property of converseness of cardinal directions in our approach. For two region objects A and B , we can use the same approach to compute $dir(A, B)$ and $dir(B, A)$. Further, we obtain the consistent result that $dir(A, B)$ is the inverse of $dir(B, A)$, and vice versa. If we have either $dir(A, B)$

or $dir(B, A)$, we can derive the inverse composite cardinal direction immediately in constant time as the union of the inverse basic cardinal directions. That is, if we know $dir(A, B)$, for example, we can immediately determine $dir(B, A)$ as $dir(B, A) = \{inv(d) \mid d \in dir(A, B)\}$ where the function inv determines the inverse of each basic cardinal direction (for example, $inv(W) = E, inv(NW) = SE$).

4. THE TILING PHASE OF THE OIM MODEL

In this section, we focus on the *tiling phase*, which is the first phase of our OIM model (see Figure 3). Section 4.1 details our novel tiling strategy that produces *objects interaction grids* of different size depending on the spatial constellation of the operand region objects. In Section 4.2, we show how objects interaction grids are represented by *objects interaction matrices* and which matrices are valid.

4.1 The Objects Interaction Grid: Capturing the Interaction of Objects

The general idea of our tiling strategy is to superimpose a grid called *objects interaction grid* (*OIG*) on a configuration of two complex region objects. Such a grid is determined by the two vertical and the two horizontal *partitioning lines* of each object. The two vertical (two horizontal) partitioning lines of an object are given as infinite extensions of the two vertical (two horizontal) segments of an object's minimum bounding rectangle. The four partitioning lines of an object create a partition of the Euclidean plane consisting of nine mutually exclusive, directional *tiles* or *zones* from which the center tile is bounded and the eight surrounding tiles are unbounded (Figure 2a). Further, these lines partition any other object into non-overlapping components such that each component is located in a different tile.

This essentially describes the tiling strategy of the Direction-Relation Matrix model. However, the fundamental difference and improvement of our model is that we apply this tiling strategy to *both* spatial operand objects, thus obtain two separate grid partitions, and then overlay both partitions (Figure 4a). This leads to an entirely novel cardinal direction model. The overlay achieves a coequal interaction and symmetric treatment of both objects. In the most general case, all partitioning lines are different from each other, and we obtain an overlay partition with 9 central, bounded tiles (indicated by the continuous segments in Figure 4a) and 16 surrounding, unbounded tiles (indicated by the dashed segments). The unbounded tiles are irrelevant for our further considerations since they do not interact with both objects. Hence, we exclude them and obtain a grid space for two region objects A and B that is bounded by the minimum bounding rectangle of the minimum bounding rectangles of A and B (Definition 4.1). This is in contrast to the partitions of all other tiling-based models that are unbounded and subdivide \mathbb{R}^2 completely.

Definition 4.1. Let $A, B \in region$ with $A \neq \emptyset$ and $B \neq \emptyset$, and let $min_x^r = \min\{x \mid (x, y) \in r\}$, $max_x^r = \max\{x \mid (x, y) \in r\}$, $min_y^r = \min\{y \mid (x, y) \in r\}$, and $max_y^r = \max\{y \mid (x, y) \in r\}$ for $r \in \{A, B\}$. Then the *objects interaction grid space* (*OIGS*) of A and B is given as

$$OIGS(A, B) = \{(x, y) \in \mathbb{R}^2 \mid \min(min_x^A, min_x^B) \leq x \leq \max(max_x^A, max_x^B) \wedge \min(min_y^A, min_y^B) \leq y \leq \max(max_y^A, max_y^B)\}$$

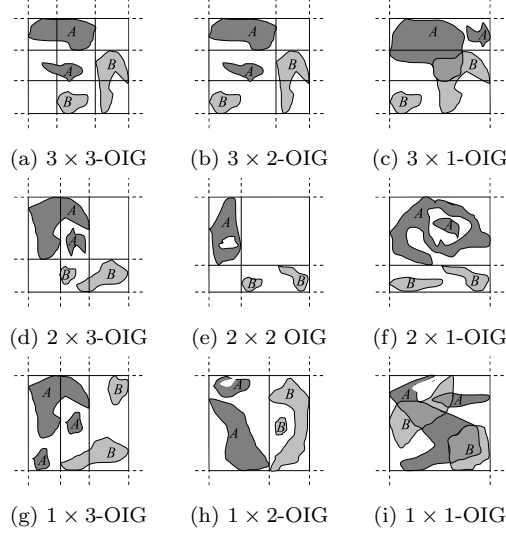


Fig. 5. Examples of the nine possible sizes of objects interaction grids (a)-(i)

Definition 4.2 determines the bounded grid formed as a part of the partitioning lines and superimposed on $OIGS(A, B)$.

Definition 4.2. Let $\min_x = \min(\min_x^A, \min_x^B)$, $\max_x = \max(\max_x^A, \max_x^B)$, $\min_y = \min(\min_y^A, \min_y^B)$, and $\max_y = \max(\max_y^A, \max_y^B)$. Let further seg be a function that constructs a segment between any two given points $p, q \in \mathbb{R}^2$, that is, $seg(p, q) = \{t \mid t = (1 - \lambda)p + \lambda q, 0 \leq \lambda \leq 1\}$. We define $H_r = \{seg((\min_x, \min_y^r), (\max_x, \min_y^r)), seg((\min_x, \max_y^r), (\max_x, \max_y^r))\}$ and $V_r = \{seg((\min_x^r, \min_y), (\min_x^r, \max_y)), seg((\max_x^r, \min_y), (\max_x^r, \max_y))\}$ for $r \in \{A, B\}$. We call the elements of H_A , H_B , V_A , and V_B *objects interaction grid segments*. Then the *objects interaction grid (OIG)* for A and B is given as

$$OIG(A, B) = H_A \cup V_A \cup H_B \cup V_B.$$

This definition implicitly comprises the specification of all grids that can arise. In the most general case, if $H_A \cap H_B = \emptyset$ and $V_A \cap V_B = \emptyset$, we obtain a bounded 3×3 -grid. Special cases arise if $H_A \cap H_B \neq \emptyset$ and/or $V_A \cap V_B \neq \emptyset$. Then equal grid segments coincide in the union of all grid segments. As a result, depending on the relative position of two region objects to each other, objects interaction grids can be of different size. However, due to the non-empty property of a region object, not all grid segments can coincide. This means that at least two horizontal grid segments and at least two vertical grid segments from two region objects must be maintained. Figure 5 shows examples for all nine possible sizes of objects interaction grids, and Definition 4.3 gives a corresponding formal characterization.

Definition 4.3. An objects interaction grid $OIG(A, B)$ is of size $m \times n$, with $m, n \in \{1, 2, 3\}$, if $|H_A \cap H_B| = 3 - m$ and $|V_A \cap V_B| = 3 - n$.

An objects interaction grid $OIG(A, B)$ of size $m \times n$, with $m, n \in \{1, 2, 3\}$, has $m + n + 2$ objects interaction grid segments. Further, $(4 \leq m + n + 2 \leq 8) \Leftrightarrow (2 \leq$

$m + n \leq 6$) holds.

An $m \times n$ -objects interaction grid partitions the objects interaction grid space into $m \cdot n$ *objects interaction grid tiles (zones, cells)* $t_{i,j}$ with $1 \leq i \leq m$ and $1 \leq j \leq n$. That is, $t_{i,j}$ is the tile at the i th row and j th column of the grid $OIG(A, B)$. All tiles are bounded, and two adjacent tiles share a common boundary. Let $OIGT(A, B)$ be the set of all tiles $t_{i,j}$ imposed by $OIG(A, B)$ on $OIGS(A, B)$.

4.2 The Objects Interaction Matrix: Representing the Interaction of Objects

By applying our tiling strategy, an objects interaction grid can be generated for any two complex (and therefore also for any two simple) region objects A and B . Such a grid provides us with the valuable information which region object intersects which tile. In Section 4.2.1, we first introduce a so-called *objects interaction function* that determines the interaction of each object with each tile of an objects interaction grid. The complete interaction of A and B with their objects interaction grid is then summarized in an *objects interaction matrix*. Since not all matrices are realizable for two complex region objects, Section 4.2.2 determines the *valid* objects interaction matrices for them. Section 4.2.3 performs the same task for two simple region objects. Cardinal directions only make sense if they are determined for valid objects interaction matrices.

4.2.1 Objects Interaction Function and Objects Interaction Matrix. Definition 4.4 provides us with a function that determines the interaction of A and B with a particular tile.

Definition 4.4. Given $A, B \in \text{region}$ with $A \neq \emptyset$ and $B \neq \emptyset$ and $OIGT(A, B)$. Let ι be the *objects interaction function* that encodes the *interaction* of A and B with a tile $t_{i,j} \in OIGT(A, B)$ and checks whether no region, A only, B only, or both regions intersect a tile. We define this function as

$$\iota(A, B, t_{i,j}) = \begin{cases} 0 & \text{if } A^\circ \cap t_{i,j}^\circ = \emptyset \wedge B^\circ \cap t_{i,j}^\circ = \emptyset \\ 1 & \text{if } A^\circ \cap t_{i,j}^\circ \neq \emptyset \wedge B^\circ \cap t_{i,j}^\circ = \emptyset \\ 2 & \text{if } A^\circ \cap t_{i,j}^\circ = \emptyset \wedge B^\circ \cap t_{i,j}^\circ \neq \emptyset \\ 3 & \text{if } A^\circ \cap t_{i,j}^\circ \neq \emptyset \wedge B^\circ \cap t_{i,j}^\circ \neq \emptyset \end{cases}$$

The operator $^\circ$ denotes the point-set topological *interior* operator and yields a region without its boundary. For each grid cell $t_{i,j}$ in the i th row and j th column of an $m \times n$ -grid with $1 \leq i \leq m$ and $1 \leq j \leq n$, we store the coded information in an *objects interaction matrix (OIM)* in cell $OIM(A, B)_{i,j}$. We thus abstract from the geometry of the objects interaction grid space and only keep the information which region intersects which tile. We can do this since directional relationships have a qualitative and not a quantitative or metric nature. Figure 6 shows the objects interaction matrix for $m = n = 3$ as an example.

$$OIM(A, B) = \begin{pmatrix} \iota(A, B, t_{1,1}) & \iota(A, B, t_{1,2}) & \iota(A, B, t_{1,3}) \\ \iota(A, B, t_{2,1}) & \iota(A, B, t_{2,2}) & \iota(A, B, t_{2,3}) \\ \iota(A, B, t_{3,1}) & \iota(A, B, t_{3,2}) & \iota(A, B, t_{3,3}) \end{pmatrix}$$

Fig. 6. The Objects Interaction Matrix of a 3×3 -Objects Interaction Grid ($\iota(A, B, t_{i,j})$ computes the interaction of A and B with tile $t_{i,j}$)

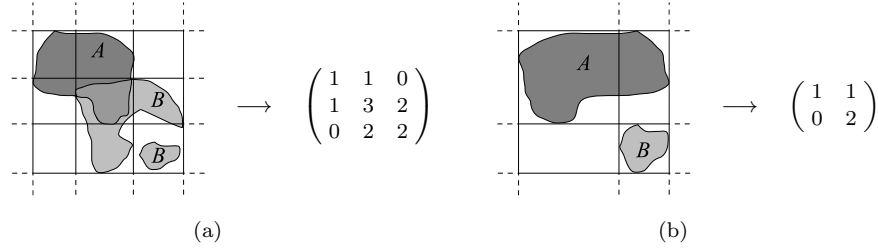


Fig. 7. Examples of mapping objects interaction grids to objects interaction matrices

Since we have nine possible $m \times n$ -configurations for objects interaction grids ($m, n \in \{1, 2, 3\}$), we have nine possible $m \times n$ -configurations for objects interaction matrices as well. Figure 7 shows two examples of mapping objects interaction grids to objects interaction matrices. For any two region objects in the plane, we can generate a unique objects interaction matrix to represent their interaction scenario. Since each matrix element may have one of the four possible values 0, 1, 2, and 3, an $m \times n$ -objects interaction matrix has 4^{mn} possible different instances. Figure 9a gives an overview of the potential numbers of matrices depending on the grid size. The total sum of matrices is

$$\sum_{i=1}^m \sum_{j=1}^n 4^{ij} = 270756$$

However, not all matrices represent valid interaction scenarios. We call an objects interaction matrix *valid* if, and only if, it can be derived from an objects interaction grid. This encoding ensures the existence of a valid interaction scenario described by the matrix. For instance, any matrix with a value 0 in all cells (Figure 8a) is invalid because both operand objects are assumed to be non-empty. Another example of an invalid matrix is any $m \times n$ matrix with $n > 1$ where the first column has the value 0 in all its cells (Figure 8b). The reason is that our tiling strategy would never produce the partitioning lines bounding the first “empty” column but let them collapse. Hence, the number of columns would decrease by one.

Open questions are now (i) which and how many matrices represent valid objects interaction matrices for two *complex* region objects (see Section 4.2.2), and (ii) which and how many matrices represent valid objects interaction matrices for two *simple* region objects (see Section 4.2.3). These two questions are important for four main reasons. First, we would like to get an indication of the expressiveness of our tiling strategy and OIM concept. At this point we know that 270756 matrices can be formed. But we do not know the possible topological structures of two complex/simple regions that represent valid spatial configurations. Second, we would

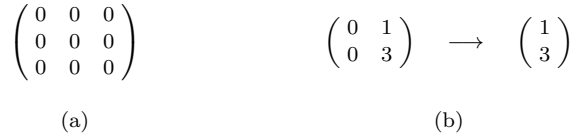


Fig. 8. Examples of invalid matrices

like to compare the number of valid objects interaction matrices with the number of valid direction-relation matrices from [Goyal and Egenhofer 2000a]. The DRM model can represent 218 valid spatial configurations for two simple regions that are represented by 218 valid direction-relation matrices which are directly mapped to 218 corresponding composite cardinal directions. Third, the interpretation function should only be applied to valid object interactions matrices that represent existing spatial real world scenarios. For invalid matrices, the interpretation function also computes a cardinal direction as a result. But this result is meaningless if the input is meaningless. This is comparable to an operation that expects correct argument objects in order to produce meaningful result objects. Incorrect argument objects would lead to incorrect results. Hence, the interpretation function can check itself (for example in an implementation) whether its input objects interaction matrix is valid. Fourth, seeing the problems of directional reasoning with the DRM model [Goyal and Egenhofer 2000a; Skiadopoulou and Koubarakis 2004], we hope that the knowledge about the valid objects interaction matrices will lead to a better understanding of directional reasoning.

4.2.2 Determining the Valid Objects Interaction Matrices for Two Complex Regions. For this purpose, we employ a three-stage proof technique called *proof-by-constraint-grouping-drawing*. In a first stage, we determine and prove the correctness of a collection of consistency constraints that can be used to filter out all impossible matrices that do hence not represent valid objects interaction matrices. We first specify some *general constraints* that are independent of any matrix size and then some *special constraints* that depend on the matrix size. In a second stage, we classify the obtained matrices from the first stage into different *groups* such that, within each group, matrices can be derived from other matrices in the same group by applying any combination of the following operations: *rotation*, *flipping*, and *role exchange*. Further, we prove that if one matrix in a group is valid, then all matrices in the same group are valid. In a third stage, we verify the correctness of all matrices by drawing prototypical spatial configurations in \mathbb{R}^2 only for the representative matrix selected from each group.

The first stage: filtering out the invalid matrices. We begin with the first stage, that is, the specification of the general constraints. Let M be a matrix of size $m \times n$ where $m, n \in \{1, 2, 3\}$. We assume that for any element $M_{i,j}$ of M , with $1 \leq i \leq m$ and $1 \leq j \leq n$, we have $M_{i,j} \in \{0, 1, 2, 3\}$. If M is a valid $m \times n$ -objects interaction matrix for two complex region objects A and B , it must satisfy the general constraints specified in Lemmas 4.5 to 4.8.

LEMMA 4.5. *At least one matrix element of M must carry the value 3, or, alternatively, at least two matrix elements of M must carry the values 1 and 2, that is,*

$$\begin{aligned} & (\exists 1 \leq i \leq m \exists 1 \leq j \leq n : M_{i,j} = 3) \vee \\ & (\exists (i,j), (i',j') \in \{1, \dots, m\} \times \{1, \dots, n\}, (i,j) \neq (i',j') : \\ & \quad M_{i,j} = 1 \wedge M_{i',j'} = 2) \end{aligned}$$

Proof. Since $A \neq \emptyset$ and $B \neq \emptyset$ holds, both region objects must intersect at least one grid cell (tile) and thus be mapped to at least one matrix element. Either they

intersect the same grid cell reflected by the value 3 in the corresponding matrix element, or they intersect two different grid cells reflected by the values 1 and 2, respectively, in the corresponding different matrix elements. \square

LEMMA 4.6. *The matrix elements of the boundary columns and the boundary rows of M are not allowed to all carry the value 0, that is,*

$$(\exists 1 \leq i \leq m \exists 1 \leq j \leq n : M_{i,1} \neq 0 \wedge M_{1,j} \neq 0) \wedge \\ (\exists 1 \leq i \leq m \exists 1 \leq j \leq n : M_{i,n} \neq 0 \wedge M_{m,j} \neq 0)$$

Proof. We assume that M is a valid objects interaction matrix and show the proof for the left boundary column of M (that is, $j = 1$). Partitioning lines are defined as extensions of the minimum bounding rectangle segments of A and B . This, in particular, implies that each partitioning line and thus each grid segment touches A and/or B in at least one point. Let us assume that the matrix elements of the left boundary column of M all carry the value 0. This means that in the objects interaction grid that has been mapped to M neither A nor B intersect any of the pertaining boundary grid cells. Consequently, the left vertical grid segment v_1 of the left boundary grid column touches neither A nor B . This is a contradiction to the definition of a partitioning line and grid segment and violates our tiling strategy. Hence, such an objects interaction grid cannot exist, and M cannot be a valid objects interaction matrix. The proofs for the right boundary column as well as for the boundary rows are similar. \square

LEMMA 4.7. *No two adjacent columns or two adjacent rows are identical in M , that is,*

$$(\forall 1 \leq i \leq m - 1 \exists 1 \leq j \leq n : M_{i,j} \neq M_{i+1,j}) \wedge \\ (\forall 1 \leq j \leq n - 1 \exists 1 \leq i \leq m : M_{i,j} \neq M_{i,j+1})$$

Proof. If there are two identical adjacent columns in M , we have to distinguish two main cases. The first case is that all matrix elements of both columns are equal to 0. But then at least one of the columns must be a boundary column since the matrices have at most three columns, and according to Lemma 4.6 at least one matrix element of a boundary column must be different from 0. The second, general case implies that, depending on the number of rows, there are at least one and at most three pairs of horizontally adjacent matrix elements with the same value unequal to 0. For each such pair, the objects on the left side and the right side of the shared and crossing partitioning line of both corresponding adjacent grid columns are the same. This contradicts to the definition of a partitioning line that the region object providing it must be located completely at either side of it. The argumentation for adjacent rows is similar. \square

The following lemmas require three auxiliary functions. The function $row(M, i) = \{M_{i,1}, \dots, M_{i,n}\}$ with $1 \leq i \leq m$ returns a set that contains all elements in row i of M . The function $col(M, j) = \{M_{1,j}, \dots, M_{m,j}\}$ with $1 \leq j \leq n$ returns a set that contains all elements in column j of M . The boundary rows for the $m \times n$ -matrix M can thus be defined as $row(M, i)$ with $i \in \{1, m\}$, and the boundary columns can be defined as $col(M, j)$ with $j \in \{1, n\}$. The function cbo , which stands for “contains both objects”, is a predicate that tests whether a row, or a column, contains parts from *both* objects. Let $S \subseteq \{0, 1, 2, 3\}$ be a set, then

$cbo(S) \Leftrightarrow 3 \in S \vee 1, 2 \in S$. Hence, the predicate $cbo(row(M, i))$ returns *true* if the i th row contains the value 3 or, if this is not the case, both values 1 and 2. Thus, for example, $cbo(\{1, 2, 3\}) = \text{true}$ and $cbo(\{1, 0, 3\}) = \text{true}$. The same applies to columns.

LEMMA 4.8. *Two adjacent columns or two adjacent rows in M that each have at least one matrix element unequal to 0 must contain a matrix element with value 3 or, if this is not the case, two matrix elements with the values 1 and 2, that is,*

$$\begin{aligned} & (\forall 1 \leq i \leq m-1 : (\exists 1 \leq k, l \leq n : M_{i,k} \neq 0 \wedge M_{i+1,l} \neq 0) \\ & \quad \Rightarrow cbo(row(M, i) \cup row(M, i+1))) \wedge \\ & (\forall 1 \leq j \leq n-1 : (\exists 1 \leq k, l \leq m : M_{k,j} \neq 0 \wedge M_{l,j+1} \neq 0) \\ & \quad \Rightarrow cbo(col(M, j) \cup col(M, j+1))) \end{aligned}$$

Proof. We assume two adjacent columns in M that both have at least one matrix element with a value unequal to 0. Further, we assume that the constraints of the lemma do not hold, that is, the two columns do neither contain a matrix element with the value 3 nor two matrix elements with the values 1 and 2. Then we can distinguish two cases. The first case is that the two adjacent columns only contain matrix elements with the values 1 or 0. This implies that no components of B lie in the two adjacent columns. Thus the partitioning line shared by the two adjacent columns does not belong to B but to A . Since each of the two adjacent columns has at least one matrix element unequal to 0, A lies on the both sides of the partitioning line. This contradicts to the definition of a partitioning line. The second case is that the two adjacent columns only contain matrix elements with the values 2 or 0. The proof is similar as before. Similarly, we can show the proof for two adjacent rows. \square

In the following Lemmas 4.9 to 4.13, we identify additional matrix-size specific constraints for $m \times n$ -objects interaction matrices of different size. We skip the formulation of constraints for 3×2 -, 3×1 -, and 2×1 -objects interaction matrices and consider their symmetric counterparts only. The reason is that for proving their correctness we can always first determine the transpose of each of these matrices and then check the transpose against one of the following matrix-specific lemmas. If the transpose fulfils (fails) the respective lemma, the original matrix fulfils (fails) it too. In case of the 1×1 -objects interaction matrix, which represents the situation that the minimum bounding rectangles of A and B are identical, the matrix element can only carry the value 3 to represent both objects. The values 0, 1, and 2 are not allowed. But this constraint is already covered by Lemma 4.5.

LEMMA 4.9. *If M is a 3×3 -objects interaction matrix, the matrix elements of each boundary column and each boundary row may neither carry the value 3 nor the two values 1 and 2, that is,*

$$(\forall i \in \{1, m\} : \neg cbo(row(M, i)) \wedge (\forall j \in \{1, n\} : \neg cbo(col(M, j)))$$

Proof. Assuming that a matrix element of a boundary column of M has the value 3, this means that a tile of the corresponding boundary grid column is intersected by both A and B . Since the tile is part of the boundary grid column, A and B would share a common partitioning line (v_1 for the left and v_4 for the right boundary grid column) and thus a common grid segment. But this is in contradiction to the

fact that a 3×3 -objects interaction matrix does not have coinciding partitioning lines. The argumentation is the same if two different matrix elements of a boundary column have the values 1 and 2 respectively. The proof is similar for boundary rows. \square

LEMMA 4.10. *If M is a 2×3 -objects interaction matrix, the matrix elements of each boundary column may neither carry the value 3 nor the two values 1 and 2, and at most one boundary row may contain a matrix element with the value 3 or, if this is not the case, two matrix elements with the values 1 and 2, that is,*

$$(\forall j \in \{1, n\} : \neg cbo(col(M, j))) \wedge (\exists i \in \{1, m\} : \neg cbo(row(M, i)))$$

Proof. The first part follows immediately from Lemma 4.9 since both matrices have the same number of columns. For the second part we assume that each of the two (boundary) rows has a matrix element with the value 3 (in the middle column), or, if this is not the case for a row, two matrix elements with the values 1 and 2. Hence, each row in the matrix contains both objects A and B . In other words, both A and B lie on both sides of the horizontal partitioning line h_2 . This leads to a contradiction since h_2 must be the partitioning line of either A or B . Thus, the object providing h_2 must be completely located at either side of it. \square

LEMMA 4.11. *If M is a 1×3 -objects interaction matrix, the matrix elements of the boundary columns may not carry the value 3, that is,*

$$M_{1,1} \neq 3 \wedge M_{1,3} \neq 3$$

Proof. This follows immediately from Lemma 4.9 since both matrices have the same number of columns. \square

LEMMA 4.12. *If M is a 2×2 -objects interaction matrix, at most one boundary row and at most one boundary column may contain a matrix element with the value 3 or, if this is not the case, two matrix elements with the values 1 and 2, that is,*

$$(\exists i \in \{1, m\} : \neg cbo(row(M, i))) \wedge (\exists j \in \{1, n\} : \neg cbo(col(M, j)))$$

Proof. If M contains more than one matrix element with the value 3, then either both columns contain a matrix element with the value 3, or both rows contain a matrix element with the value 3. This means that A and B lie on both sides of the vertical partitioning line v_2 shared by both columns, or on both sides of the horizontal partitioning line h_2 shared by both rows. This contradicts the definition of a partitioning line since v_2 and h_2 respectively can be a partitioning line of neither A nor B . \square

LEMMA 4.13. *If M is a 1×2 -objects interaction matrix, at most one matrix element may carry the value 3, that is,*

$$M_{1,1} \neq 3 \vee M_{1,2} \neq 3$$

Proof. The argument is the same as in Lemma 4.12 with respect to columns. \square

So far, in this first stage, we have defined a number of consistency constraints for removing invalid $m \times n$ -matrices that cannot be valid objects interaction matrices for two complex region objects. In particular, we have shown the soundness and correctness of these constraints. We have developed a simple program that checks

		n		
m	\times	1	2	3
	1	4	16	64
	2	16	256	4096
	3	64	4096	262144
Total: 270756				
(a)				

		n		
m	\times	1	2	3
	1	1/1	6/6	8/6
	2	6/6	84/68	216/124
	3	8/6	216/124	1132/464
Total: 1677/805				
(b)				

Matrices of Size(s)	# Matrix Groups
1×1	1/1
1×2 and 2×1	2/2
1×3 and 3×1	4/3
2×2	9/7
2×3 and 3×2	34/20
3×3	115/49
Total: 165/82	
(c)	

Fig. 9. Numbers of possible $m \times n$ -matrices in general (a), numbers of valid objects interaction matrices for two complex/simple region objects (b), and numbers of matrix groups with respect to validity testing for two complex/simple region objects (c) depending on the grid/matrix size and in total

the validity of all 270756 $m \times n$ -matrices (see Figure 9a) with respect to the general and special constraints of the Lemmas 4.5 to 4.13. The result is shown in Figure 9b and indicates the large reduction of the number of matrices to 1677. But still, due to space limitations, we cannot show these matrices. We order them so that each matrix obtains a unique *matrix number*. The order is given by the matrix size (first row, then column) and within each size by the increasing 9-digit number to the base 4 obtained as the concatenation of the rows and possibly filled up with zeros at the beginning. This leads to one 1×1 -matrix with matrix number 1, 6 1×2 -matrices with the matrix numbers 2 to 7, 8 1×3 -matrices with the matrix numbers 8 to 15, 6 2×1 -matrices with the matrix numbers 16 to 21, 84 2×2 -matrices with the matrix numbers 22 to 105, 216 2×3 -matrices with the matrix numbers 106 to 321, 8 3×1 -matrices with the matrix numbers 322 to 329, 216 3×2 -matrices with the matrix numbers 330 to 545, and 1132 3×3 -matrices with the matrix numbers 546 to 1677.

The second stage: classifying the remaining matrices. Our remaining task is to show the completeness of the constraints (that is, no further constraint is missing) and the correctness of the remaining matrices (that is, the remaining matrices are valid). For this purpose, the general idea of the proof-by-constraint-grouping-drawing technique is to show for each remaining matrix that a spatial configuration of two spatial objects A and B together with their corresponding objects interaction grid exists which represents a real world scenario and can be mapped to the matrix under consideration. However, the creation of spatial configuration drawings for all 1677 matrices is a tedious effort. Hence, we perform a preprocessing step before and, in a second stage, classify the remaining matrices into different *groups* (equivalence classes) such that, within each group, all matrices can be derived from the other matrices in the same group by applying any combination of the operations *rotation*,

horizontal flipping, vertical flipping, and role exchange. We formally define these four operations and then prove for each operation that its application to a valid matrix results in a new valid matrix. A recursive application of these operations then means that if one matrix in a group is valid, then all matrices in the same group are valid. Finally, we provide an algorithm that is based on the recursive application of these four operations and partitions all 1677 matrices into groups. The advantage of this preprocessing step is that in the third stage discussed below we only have to draw a spatial configuration and an objects interaction grid for one selected representative of each group. This leads to a large reduction of drawings, as we will see.

Definition 4.14 specifies the operation *rota* that rotates any $m \times n$ -matrix by 90° around its right lowermost matrix element. Figure 10a shows an illustration of this operation; the rotation matrix element is v_9 . Note that for $m \neq n$ the matrix size changes. For example, a 1×3 -matrix is rotated into a 3×1 -matrix.

Definition 4.14. Let M be an $m \times n$ -matrix and M' be the $n \times m$ -matrix that we obtain by a clockwise rotation of M by 90° . The rotation operation *rota* is then defined as follows:

$$M' = \text{rota}(M) \stackrel{\text{def}}{\iff} \forall 1 \leq i \leq m \forall 1 \leq j \leq n : M'_{j,m+1-i} = M_{i,j}$$

While Definition 4.14 holds for arbitrary matrices, Lemma 4.15 shows that the application of the rotation operation to a valid objects interaction matrix amounts to a new valid objects interaction matrix.

LEMMA 4.15. *Let M be a valid objects interaction matrix. Then the matrix M' obtained by applying the operation *rota* to M is also a valid objects interaction matrix.*

Proof. Since, according to our assumption, M is a valid objects interaction matrix of size $m \times n$, two region objects A and B and their objects interaction grid $G = OIG(A, B)$ of size $m \times n$ must exist such that $M = OIM(A, B)$ is their pertaining objects interaction matrix. A rotation is an isometry or congruence mapping of the plane, that is, a linear transformation which preserves length, distance, shape, and size of spatial objects. Hence, if we rotate A and B clockwise by 90° , we obtain congruent region objects A' and B' respectively, and their objects interaction grid $G' = OIG(A', B')$ is congruent to G , obtained by a clockwise rotation of G by 90° , and of size $n \times m$. This means that for all $1 \leq i \leq m$ and for all $1 \leq j \leq n$ each tile $t_{i,j} \in OIGT(A, B)$ with the corresponding object parts of A and B is rotated and mapped to a tile $t_{j,m+1-i} \in OIGT(A', B')$ with the corresponding rotated object parts of A' and B' . This rotated spatial configuration exists and can therefore be mapped to a valid objects interaction matrix M' by a clockwise rotation of M by 90° . This clockwise rotation is obtained by the assignment that $\iota(A', B', t_{j,m+1-i})$ is set to $\iota(A, B, t_{i,j})$, that is, a matrix rotation is ι -preserving. \square

Definition 4.16 defines the two operations *hflip* for horizontal flipping and *vflip* for vertical flipping that mirror any matrix along a vertical axis and a horizontal axis respectively. Figures 10b and 10c illustrate both operations.

Definition 4.16. Let M be an $m \times n$ -matrix and M' and M'' be $m \times n$ -matrices that we obtain by flipping M with respect to its vertical axis and horizontal axis re-

$$\begin{array}{ccc}
\begin{pmatrix} v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 \end{pmatrix} & \xrightarrow{\text{rotation}} & \begin{pmatrix} v_7 & v_4 & v_1 \\ v_8 & v_5 & v_2 \\ v_9 & v_6 & v_3 \end{pmatrix} & \begin{pmatrix} v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 \end{pmatrix} & \xrightarrow[\text{flipping}]{\text{horizontal}} & \begin{pmatrix} v_3 & v_2 & v_1 \\ v_6 & v_5 & v_4 \\ v_9 & v_8 & v_7 \end{pmatrix} \\
\text{(a)} & & & \text{(b)}
\end{array}$$

$$\begin{array}{ccc}
\begin{pmatrix} v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 \end{pmatrix} & \xrightarrow[\text{flipping}]{\text{vertical}} & \begin{pmatrix} v_7 & v_8 & v_9 \\ v_4 & v_5 & v_6 \\ v_1 & v_2 & v_3 \end{pmatrix} & \begin{pmatrix} v_1 & v_2 & v_3 \\ v_4 & v_5 & v_6 \\ v_7 & v_8 & v_9 \end{pmatrix} & \xrightarrow[\text{exchange}]{\text{role}} & \begin{pmatrix} \overline{v_1} & \overline{v_2} & \overline{v_3} \\ \overline{v_4} & \overline{v_5} & \overline{v_6} \\ \overline{v_7} & \overline{v_8} & \overline{v_9} \end{pmatrix} \\
\text{(c)} & & & \text{(d)}
\end{array}$$

Fig. 10. The operations *rotation* (a), *horizontal flipping* (b), *vertical flipping* (c), and *role-exchange* ($\overline{v_i} = \text{if } v_i \in \{0, 3\} \text{ then } v_i \text{ else if } v_i = 1 \text{ then } 2 \text{ else } 1$) (d).

spectively. The operations *hflip* for horizontal flipping and *vflip* for vertical flipping are then defined as follows:

$$\begin{aligned}
M' &= \text{hflip}(M) \stackrel{\text{def}}{\iff} \forall 1 \leq i \leq m \ \forall 1 \leq j \leq n : M'_{i,n+1-j} = M_{i,j} \\
M'' &= \text{vflip}(M) \stackrel{\text{def}}{\iff} \forall 1 \leq i \leq m \ \forall 1 \leq j \leq n : M''_{m+1-i,j} = M_{i,j}
\end{aligned}$$

Lemma 4.17 shows that both flipping operations map a valid objects interaction matrix into a new valid objects interaction matrix.

LEMMA 4.17. *Let M be a valid objects interaction matrix. Then the matrices M' and M'' obtained by applying the operations *hflip* and *vflip* respectively to M are also valid objects interaction matrices.*

Proof. The argumentation is very similar to the proof of Lemma 4.15 since horizontal flipping and vertical flipping are also isometries, that is, congruence mappings, in the plane. The main differences consist in the preservation of the matrix size of $m \times n$ and in the manner how tiles are exchanged. For horizontal (vertical) flipping, for all $1 \leq i \leq m$ and for all $1 \leq j \leq n$, a tile $t_{i,j} \in \text{OIGT}(A, B)$ with the corresponding object parts of A and B is flipped to a tile $t_{i,n+1-j} \in \text{OIGT}(A', B')$ ($t_{m+1-i,j} \in \text{OIGT}(A', B')$) with the corresponding flipped object parts of A' and B' . Thus, the horizontally and vertically flipped spatial configurations exist and can be mapped to valid objects interaction matrices M' and M'' respectively that are obtained by horizontal flipping and vertical flipping of M . The horizontal (vertical) flipping of M to M' (M'') is obtained by the assignment that $\iota(A', B', t_{i,n+1-j})$ ($\iota(A', B', t_{m+1-i,j})$) is set to $\iota(A, B, t_{i,j})$. Hence, flipping is ι -preserving too. \square

Definition 4.18 specifies the operation *role_ex* that swaps the spatial object names A and B and thus their encodings 1 and 2 and is applied to one of the remaining 1677 matrices. Figure 10d illustrates this operation.

Definition 4.18. Let M be one of the 1677 $m \times n$ -matrices and M' be the $m \times n$ -matrix that we obtain by changing the roles of the two interacting spatial objects A and B in M . The operation *role_ex* is then defined as follows:

$$M' = \text{role_ex}(M) \stackrel{\text{def}}{\iff} \forall 1 \leq i \leq m \ \forall 1 \leq j \leq n : M'_{i,j} = \begin{cases} 0 & \text{if } M_{i,j} = 0 \\ 1 & \text{if } M_{i,j} = 2 \\ 2 & \text{if } M_{i,j} = 1 \\ 3 & \text{if } M_{i,j} = 3 \end{cases}$$

Lemma 4.19 shows that swapping the roles of A and B in a valid objects interaction matrix results in a new valid objects interaction matrix.

LEMMA 4.19. *Let M be a valid objects interaction matrix. Then the matrix M' obtained by applying the operation `role_ex` to M is also a valid objects interaction matrix.*

Proof. Since M is a valid objects interaction matrix, two region objects A and B and their objects interaction grid $G = OIG(A, B)$ must exist such that $M = OIM(A, B)$ is their objects interaction matrix. The role-exchange operation does not change or even invalidate a spatial configuration since only region object names are exchanged while the geometries remain unaltered. Due to the preservation of the object geometries, a role exchange does also not have influence on the objects interaction grid which remains the same. Thus, the spatial configuration after a role exchange exists and can therefore be mapped to a valid objects interaction matrix M' by exchanging the matrix values 1 (encoding for A) and 2 (encoding for B) in M . The matrix elements 0 and 3 remain unaltered. \square

In summary, we have shown in Lemmas 4.15 to 4.19 that the application of one of the operations `rota`, `hflip`, `vflip`, and `role_ex` to a valid objects interaction matrix yields a new valid objects interaction matrix. Lemma 4.20 makes the conclusion that any composition (iterative application) of these operations with respect to a valid objects interaction matrix leads to a valid objects interaction matrix.

LEMMA 4.20. *Let M be a valid objects interaction matrix. The composition of the operations `rota`, `hflip`, `vflip`, and `role_ex` to M in any order and with any quantity yields a valid objects interaction matrix M' .*

Proof. Let $n \in \mathbb{N}$ be an arbitrary natural number, and let $\omega_1, \omega_2, \dots, \omega_n \in \{\text{rota}, \text{hflip}, \text{vflip}, \text{role_ex}\}$. Further, let $M' = \omega_n(\omega_{n-1}(\dots(\omega_2(\omega_1(M)))\dots))$ be the (function) composition of $\omega_1, \omega_2, \dots, \omega_n$ applied to any valid objects interaction matrix M . This means we obtain intermediate matrices $M_{\alpha_1}, M_{\alpha_2}, \dots, M_{\alpha_{n-1}}$ with $\alpha_1, \dots, \alpha_{n-1} \in \{1, \dots, 1677\}$ as follows:

$$M \xrightarrow{\omega_1} M_{\alpha_1} \xrightarrow{\omega_2} M_{\alpha_2} \xrightarrow{\omega_3} \dots \xrightarrow{\omega_{n-1}} M_{\alpha_{n-1}} \xrightarrow{\omega_n} M'$$

According to the Lemmas 4.15 to 4.19, M_{α_1} must be a valid objects interaction matrix. But if M_{α_1} is a valid objects interaction matrix, then, due to the same lemmas, M_{α_2} must be a valid objects interaction matrix too, and so forth. That is, all matrices $M_{\alpha_1}, M_{\alpha_2}, \dots, M_{\alpha_{n-1}}$ and finally the target M' are valid objects interaction matrices. \square

Lemma 4.21 forms the basis for our algorithm below to partition all 1677 matrices into disjoint groups.

LEMMA 4.21. *The recursive application of the operations `rota`, `hflip`, `vflip`, and `role_ex` to all 1677 remaining matrices generates an exhaustive and finite partition of these matrices into disjoint, finite objects interaction matrix groups.*

Proof. Our argumentation is constructive: We take repeatedly one of the still unconsidered 1677 matrices and apply the four operations `rota`, `hflip`, `vflip`, and `role_ex` to it. For each new matrix obtained we first check whether this matrix has already been generated before and thus already belongs to a matrix group. If this

$$\begin{aligned}
M_{595} &= \begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 0 & 0 \end{pmatrix} & M_{694} &= \begin{pmatrix} 0 & 0 & 2 \\ 2 & 3 & 2 \\ 2 & 0 & 0 \end{pmatrix} & M_{833} &= \begin{pmatrix} 0 & 1 & 1 \\ 0 & 3 & 0 \\ 1 & 1 & 0 \end{pmatrix} & M_{1039} &= \begin{pmatrix} 0 & 2 & 2 \\ 0 & 3 & 0 \\ 2 & 2 & 0 \end{pmatrix} \\
M_{1168} &= \begin{pmatrix} 1 & 0 & 0 \\ 1 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix} & M_{1266} &= \begin{pmatrix} 1 & 1 & 0 \\ 0 & 3 & 0 \\ 0 & 1 & 1 \end{pmatrix} & M_{1460} &= \begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 2 \\ 0 & 0 & 2 \end{pmatrix} & M_{1553} &= \begin{pmatrix} 2 & 2 & 0 \\ 0 & 3 & 0 \\ 0 & 2 & 2 \end{pmatrix}
\end{aligned}$$

Fig. 11. An example of an objects interaction matrix group (subscripts are matrix numbers)

is the case, the recursion for this matrix and search path ends here. Otherwise, we assign this matrix to the group under construction and recursively repeat this process and apply the four operations and the matrix checks again. At some point, this whole recursive process will terminate and lead to a group with finitely many matrices since the total number of matrices is bounded by 1677. If a group has been determined, the recursive process is repeatedly applied to another unconsidered matrix. In total, we obtain a finite number of matrix groups due to the finite number of available matrices. Further, all matrices are considered since the method takes into account all 1677 matrices and is thus exhaustive. Different matrix groups contain different matrices and are thus disjoint. Assume that this is not the case and that two different groups share exactly one matrix. Then the shared matrix would generate the matrices of both groups so that both groups would have to be merged into one group. But this is a contradiction to our assumption of two different groups. \square

An example of an objects interaction matrix group that consists of eight matrices is shown in Figure 11. Beginning with any of these eight matrices and by recursively applying the four operations *rota*, *hflip*, *vflip*, and *role_ex*, we can generate the other seven matrices in the matrix group. Other matrices cannot be generated. For instance, beginning with matrix M_{595} , the other seven matrices can be generated as follows:

$$\begin{aligned}
M_{694} &= \text{role_ex}(M_{595}) & M_{833} &= \text{hflip}(\text{rota}(M_{595})) \\
M_{1039} &= \text{role_ex}(\text{hflip}(\text{rota}(M_{595}))) & M_{1168} &= \text{hflip}(M_{595}) \\
M_{1266} &= \text{rota}(M_{595}) & M_{1460} &= \text{rota}(\text{hflip}(\text{rota}(M_{595}))) \\
M_{1553} &= \text{role_ex}(\text{rota}(M_{595}))
\end{aligned}$$

We now present an algorithm (Algorithm 1) that according to Lemma 4.21 partitions the 1677 matrices for two complex region objects into matrix groups and that determines a representative matrix for each group. The algorithm takes an array ML as input which keeps 1677 records and whose index range represents the matrix numbers 1 to 1677 (line 1). Each record, called *matrix description*, stores an objects interaction matrix *oim*, an identifier *groupid* of the group to which the matrix belongs, a Boolean flag *repr* that indicates whether the matrix *oim* is a representative of a matrix group, and the size of the matrix which is not needed in the algorithm but later used in Figure 9c to distinguish the matrix groups by matrix sizes. At the beginning, in all matrix descriptions, the group identifiers are initialized with 0 indicating that no group has been assigned so far, and the Boolean flags *repr* are initialized with *false* (line 2). A FIFO queue MQ is used as the underlying data structure to hold all matrices that need to be further examined.

Algorithm 1: Computation of the objects interaction matrix groups and their representative matrices

Input: An array ML of 1677 records. Each index between 1 and 1677 is a matrix number. Each record stores an objects interaction matrix oim , a so far non-assigned $groupid$ to which the matrix belongs, and a so far non-assigned Boolean value $repr$ marking this matrix as a representative of a matrix group

Output: The array ML with $groupid$ and $repr$ values for each matrix record determined by a breadth-first-search (BFS) strategy. As a side-effect, the total number of matrix groups as well as the representative matrix of each matrix group are printed out.

```

1  $i \leftarrow 1$ ;  $n \leftarrow 1677$ ;  $gid \leftarrow 0$ ;
2 foreach  $i$  in  $1 \dots n$  do  $ML[i].groupid \leftarrow 0$ ;  $ML[i].repr \leftarrow false$ ;
3 while  $i \leq n$  do
4   if  $ML[i].groupid = 0$  then // A new matrix group with the representative matrix  $i$ 
5      $gid \leftarrow gid + 1$ ;  $ML[i].groupid \leftarrow gid$ ;  $ML[i].repr \leftarrow true$ ;
6      $MQ \leftarrow empty()$ ;  $MQ \leftarrow enqueue(MQ, ML[i].oim)$ ;
7     while not  $isempty(MQ)$  do // Apply a BFS strategy beginning with matrix  $i$ 
8        $m \leftarrow dequeue(MQ)$ ;
9        $m_{rota} \leftarrow rota(m)$ ;  $m_{hflip} \leftarrow hflip(m)$ ;  $m_{vflip} \leftarrow vflip(m)$ ;  $m_{role\_ex} \leftarrow role\_ex(m)$ ;
10      if  $unconsidered(ML, m_{rota}, matrix\_num)$  then
11         $ML[matrix\_num].groupid \leftarrow gid$ ;  $MQ \leftarrow enqueue(MQ, m_{rota})$ ;
12      if  $unconsidered(ML, m_{hflip}, matrix\_num)$  then
13         $ML[matrix\_num].groupid \leftarrow gid$ ;  $MQ \leftarrow enqueue(MQ, m_{hflip})$ ;
14      if  $unconsidered(ML, m_{vflip}, matrix\_num)$  then
15         $ML[matrix\_num].groupid \leftarrow gid$ ;  $MQ \leftarrow enqueue(MQ, m_{vflip})$ ;
16      if  $unconsidered(ML, m_{role\_ex}, matrix\_num)$  then
17         $ML[matrix\_num].groupid \leftarrow gid$ ;  $MQ \leftarrow enqueue(MQ, m_{role\_ex})$ ;
18    $i \leftarrow i + 1$ ;
19 println( $gid$ ; " matrix groups have been identified.");
20 foreach  $i$  in  $1 \dots n$  do
21   if  $ML[i].repr$  then
22     println("Matrix no. ";  $i$ ; " is the representative of matrix group ";  $ML[i].groupid$ );
23 return  $ML$ ;

```

It offers the operations *empty* to create a new queue, *enqueue* to append a matrix to the queue, *dequeue* to access and remove the first matrix of a queue, and *isempty* to test whether a queue is empty.

Our algorithm consecutively considers all 1677 matrices (lines 3 and 18). If a group identifier has already been assigned to the matrix description $ML[i]$, this matrix description is skipped, and the next matrix description $ML[i + 1]$ is considered. Otherwise, if a group identifier has not yet been assigned to this matrix description, that is, the current group identifier is 0 (line 4), we have found the representative matrix of a new matrix group to be determined. The reason is that we select that matrix of a matrix group as the representative matrix for this group that has the lowest matrix number. Therefore, we increment our currently highest matrix group number gid , assign the updated group number to the matrix description $ML[i]$, and mark this matrix as a representative matrix by setting its Boolean value $repr$ to *true* (line 5). Next, we have to find all matrices that belong to the same group as the matrix oim of $ML[i]$. We first create a new FIFO queue MQ and initialize it

with the matrix *oim* of $ML[i]$ (line 6). Afterwards we apply a breadth-first search (BFS) as our algorithmic strategy (lines 7 to 17). As long as the queue MQ is not empty, we remove a matrix from its beginning (line 8), apply the four operations *rota*, *hflip*, *vflip* and *role_ex* to this matrix, and obtain the generated matrices m_{rota} , m_{hflip} , m_{vflip} , and m_{role_ex} (line 9). For all four matrices we perform the same actions. The Boolean predicate *unconsidered* checks for each matrix whether it has already been considered before and thus already belongs to the same group (lines 10, 12, 14, 16). If this is the case, the predicate yields *false*, and this search path is pruned. Otherwise, as a side-effect, the predicate returns the corresponding matrix number in the variable *matrix_num*, and the matrix with this matrix number obtains the current group identifier *gid*. In addition, this matrix is appended to MQ for further exploration (lines 11, 13, 15, 17). Finally, for each representative matrix of a group we have performed a breadth-first search. The termination of each search is indicated by an empty queue MQ . At the end of the algorithm we have determined all matrix groups by labeling each matrix accordingly. Besides returning the updated array ML of matrix descriptions (line 23), as a side effect of the algorithm, we print out the number of matrix groups found (line 19) and further output which matrix is the representative matrix of which matrix group (lines 20 to 22).

For reasons of clarity and simplicity, we have omitted the presentation of a number of consistency checks. These consistency checks test, for example, whether a matrix generated by the four operations *rota*, *hflip*, *vflip* and *role_ex* can be actually found in ML . We did not find any generated matrix outside of ML . Further, another consistency check tests whether a generated matrix that is found in ML with a group identifier unequal to 0 has a group identifier that is also unequal to the current group identifier *gid*. This would mean that a matrix belongs to two different groups. But this is excluded by Lemma 4.21. Indeed, we did not find any generated matrix assigned to two or more matrix groups.

The application of this algorithm leads to the result presented in Figure 9c for the different matrix sizes. We observe that several matrix groups contain matrices of different matrix size. The reason is the rotation operation that transforms an $m \times n$ -matrix into an $n \times m$ -matrix. This means that for $m \neq n$ the matrix size changes. Further, for all matrix groups that contain $m \times n$ -matrices and $n \times m$ -matrices with $m < n$, the representation matrix is always an $m \times n$ -matrix since $m \times n$ -matrices have a smaller order than $n \times m$ -matrices. In total, we obtain 165 matrix groups for the case of two complex region objects.

The third stage: verifying the correctness of the remaining matrices. In the third and last stage of our *proof-by-constraint-grouping-drawing* technique, we have to show the completeness of the constraints and the correctness of the remaining 1677 objects interaction matrices. We perform the validation of all matrices by drawing prototypical spatial configurations in \mathbb{R}^2 only for the representative matrices selected from the 165 matrix groups. If we can show that the representative matrix of a given matrix group is valid, that is, if we can show that a realistic spatial configuration with two region objects A and B as well as a corresponding objects interaction grid exists that can be mapped to the objects interaction matrix under consideration, we know that all matrices of this group are valid. This

follows from the Lemmas 4.20 and 4.21. In the electronic online appendix, we show representative objects interaction matrices and prototypical spatial configurations with their objects interaction grids for all 165 matrix groups. For each representative matrix, we provide its matrix number, the identifier of the group to which the matrix belongs, the total number of matrices that belong to this matrix group, and whether the matrix is only valid for complex regions or also valid for simple regions. We finally obtain the following result in Theorem 4.22:

THEOREM 4.22. *A matrix M is a valid objects interaction matrix of two complex region objects A and B if, and only if, all general constraints of the Lemmas 4.5 to 4.8 and all special constraints of the Lemmas 4.9 to 4.13 are fulfilled by M .*

4.2.3 Determining the Valid Objects Interaction Matrices for Two Simple Regions. Due to their simple structure, *simple* region objects have always attracted particular interest among researchers in spatial information science. As we have seen in Section 2, most available cardinal direction models are based on simple spatial objects. Although our model is more general, we now consider the case of simple regions as a special case of complex regions in order to be able to compare our approach to others.

Beside the constraints in Lemmas 4.5 to 4.13, an additional constraint identifies those matrices among the 1677 objects interaction matrices for two complex region objects that are also valid for two simple region objects. A simple region object (see Figure 1c) is defined as a bounded, regular closed set homeomorphic (that is, topologically equivalent) to a closed disc in \mathbb{R}^2 [Schneider and Behr 2006]. This, in particular, means that it has a connected interior, a connected boundary, and a connected exterior. Hence, it is not allowed to consist of several components, and it must not have holes.

The added constraint is given by the additional characteristic property of a simple region object that it is *4-connected* in an objects interaction grid. Intuitively, this means that it can only intersect grid cells that are horizontally or vertically adjacent to at least one other grid cell that it intersects. Transferred to a corresponding objects interaction matrix, this means that for a simple region object A (B), each matrix element with a value 1 (2) or 3 can only have horizontally or vertically adjacent matrix elements with the value 1 (2) or 3. Definition 4.23 defines the concept of 4-connectedness in an $m \times n$ -matrix.

Definition 4.23. Let M be an $m \times n$ -objects interaction matrix for two complex region objects A and B , and let $C_M = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. We call the values of C_M *matrix element positions*. Two matrix element positions (i, j) and (k, l) are *4-adjacent with respect to A (B)* if, and only if, $((k = i \wedge |l - j| = 1) \vee (|k - i| = 1 \wedge l = j)) \wedge M_{i,j}, M_{k,l} \in \{1, 3\} (\{2, 3\})$. A *4-path* in C_M *with respect to A (B)* is a sequence of matrix element positions m_1, \dots, m_k with $2 \leq k \leq mn$ such that m_i and m_{i+1} are 4-adjacent with respect to A (B) for all $1 \leq i < k$. A set $C \subseteq C_M$ is *4-connected with respect to A (B)* if for every pair m_p, m_q of matrix element positions in C there is a 4-path from m_p to m_q in C with respect to A (B).

Lemma 4.24 applies Definition 4.23 to identify those objects interaction matrices that are also valid for two simple region objects.

LEMMA 4.24. *Let M be one of the 1677 $m \times n$ -objects interaction matrices for two complex region objects A and B . Further, let $C_A = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n, M_{i,j} = 1 \vee M_{i,j} = 3\}$ and $C_B = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n, M_{i,j} = 2 \vee M_{i,j} = 3\}$. If A and B are simple region objects, then C_A must be 4-connected with respect to A , and C_B must be 4-connected with respect to B .*

Proof. If C_A is not a 4-connected set, it must contain two matrix element positions that do not have a 4-path connecting them. Two cases are possible. The first case is that there is not *any* path at all that connects the two positions. This indicates that A has a disconnected interior and a disconnected boundary, which contradicts the definition of a simple region object. The second case is that there is a path connecting the two matrix element positions but the path involves two consecutive positions that are diagonally adjacent. This implies that the two positions are connected at a common corner point only. But this means that A 's interior is disconnected and contains two components that share exactly one common single point. This also contradicts the definition of a simple region object. Hence, C_A must be 4-connected with respect to A . The proof for C_B is similar. \square

We finally obtain the following result in Theorem 4.25:

THEOREM 4.25. *A matrix M is a valid objects interaction matrix of two simple region objects A and B if, and only if, M is a valid objects interaction matrix of two complex regions and M also satisfies Lemma 4.24.*

The numbers right of the slashes in Figure 9b reflect the effect of applying the two additional constraints of connected interiors, connected boundaries, and connected exteriors as well as 4-connectedness to the 1677 objects interaction matrices for two complex region objects. The total number of valid objects interaction matrices for two simple region objects is 805, which is a reduction of more than 50%. Figure 9c shows that the number of matrix groups for two simple region objects is 82, which is also a reduction of more than 50%. In the electronic online appendix, all representative matrices that are valid for two simple region objects are marked with a flag *s*. A comparison of the 805 valid objects interaction matrices with the 218 valid direction-relation matrices [Goyal and Egenhofer 2000a] for two simple region objects is problematic since the underlying tiling strategies and the underlying semantics of the two kinds of matrices are different. However, we can say that the OIM model provides a much more fine-grained and complete identification of the possible valid spatial configurations between two simple regions than the DRM model.

5. THE INTERPRETATION PHASE OF THE OIM MODEL

The second phase of the OIM model is the *interpretation phase* (see Figure 3). This phase takes an objects interaction matrix obtained as the result of the tiling phase as input and uses it to generate a set of basic cardinal directions as output. This is achieved by separately identifying the locations of both objects in the objects interaction matrix and by pairwise interpreting these locations in terms of basic cardinal directions. The union of all these basic cardinal directions is the result.

In a first step, we define a function *loc* (see Definition 5.1) that acts on one of the region objects A or B and their common objects interaction matrix $OIM(A, B)$ and

determines all locations of components of each object in the matrix. Let $I_{m,n} = \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. We use an index pair $(i, j) \in I_{m,n}$ to represent the location of the element $M_{i,j} \in \{0, 1, 2, 3\}$ and thus the location of an object component from A or B in an $m \times n$ -objects interaction matrix.

Definition 5.1. Let $M = OIM(A, B)$ be the $m \times n$ -objects interaction matrix of two region objects A and B . Then the function loc is defined as:

$$\begin{aligned} loc(A, M) &= \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n, M_{i,j} = 1 \vee M_{i,j} = 3\} \\ loc(B, M) &= \{(i, j) \mid 1 \leq i \leq m, 1 \leq j \leq n, M_{i,j} = 2 \vee M_{i,j} = 3\} \end{aligned}$$

For example, in Figure 4b, object A occupies the locations (2,2) and (3,2), and object B occupies the locations (1,1) and (2,3) in the objects interaction matrix $OIM(A, B)$. Therefore, we obtain $loc(A, OIM(A, B)) = \{(2, 2), (3, 2)\}$ and $loc(B, OIM(A, B)) = \{(1, 1), (2, 3)\}$.

In a second step, we define an *interpretation function* ψ to determine the cardinal direction between any two object components of A and B on the basis of their locations in the objects interaction matrix. We use a popular model with the nine basic cardinal directions *north* (N), *northwest* (NW), *west* (W), *southwest* (SW), *south* (S), *southeast* (SE), *east* (E), *northeast* (NE), and *origin* (O) to symbolize the possible cardinal directions between *object components*. In summary, we obtain the set $CD = \{N, NW, W, SW, S, SE, E, NE, O\}$ of *basic cardinal directions*. Definition 5.2 provides the interpretation function ψ with the signature $\psi : I_{m,n} \times I_{m,n} \rightarrow CD$.

Definition 5.2. Given $(i, j), (i', j') \in I_{m,n}$, the *interpretation function* ψ on the basis of the set $CD = \{N, NW, W, SW, S, SE, E, NE, O\}$ of *basic cardinal directions* is defined as

$$\psi((i, j), (i', j')) = \begin{cases} N & \text{if } i < i' \wedge j = j' \\ NW & \text{if } i < i' \wedge j < j' \\ W & \text{if } i = i' \wedge j < j' \\ SW & \text{if } i > i' \wedge j < j' \\ S & \text{if } i > i' \wedge j = j' \\ SE & \text{if } i > i' \wedge j > j' \\ E & \text{if } i = i' \wedge j > j' \\ NE & \text{if } i < i' \wedge j > j' \\ O & \text{if } i = i' \wedge j = j' \end{cases}$$

For example, in Figure 4b, we obtain that $\psi((3, 2), (1, 1)) = SE$ and $\psi((2, 2), (2, 3)) = W$ where holds that $(2, 2), (3, 2) \in loc(A, OIM(A, B))$ and $(1, 1), (2, 3) \in loc(B, OIM(A, B))$. Table II called *interpretation table* shows the possible results of the interpretation function for all index pairs. We observe two fundamental properties in this table:

$$\begin{aligned} \forall (i, j), (i', j') \in I_{m,n} : & \text{(a) } (i, j) = (i', j') \Rightarrow \psi((i, j), (i', j')) = O \\ & \text{(b) } \psi((i, j), (i', j')) = inv(\psi((i', j'), (i, j))) \end{aligned}$$

The first property maps the cardinal direction of object components of A and B that are located in the same tile to the origin direction. The second property

$(i,j) \backslash (i',j')$	(1,1)	(1,2)	(1,3)	(2,1)	(2,2)	(2,3)	(3,1)	(3,2)	(3,3)
(1,1)	O	W	W	N	NW	NW	N	NW	NW
(1,2)	E	O	W	NE	N	NW	NE	N	NW
(1,3)	E	E	O	NE	NE	N	NE	NE	N
(2,1)	S	SW	SW	O	W	W	N	NW	NW
(2,2)	SE	S	SW	E	O	W	NE	N	NW
(2,3)	SE	SE	S	E	E	O	NE	NE	N
(3,1)	S	SW	SW	S	SW	SW	O	W	W
(3,2)	SE	S	SW	SE	S	SW	E	O	W
(3,3)	SE	SE	S	SE	SE	S	E	E	O

Table II. Interpretation table for the interpretation function ψ

makes use of a converseness function $inv : CD \rightarrow CD$ and reflects the converseness of each basic cardinal direction (for example, $inv(W) = E, inv(NW) = SE$).

In a third and final step, we specify a *cardinal direction function* named *dir* which determines the *composite cardinal direction* for two region objects A and B . This function has the signature $dir : region \times region \rightarrow 2^{CD}$ and yields a set of basic cardinal directions as its result. In order to be able to define the function *dir*, we first generalize the signature of our interpretation function ψ to $\psi : 2^{I_{m,n}} \times 2^{I_{m,n}} \rightarrow 2^{CD}$ such that for any two sets $X, Y \subseteq I_{m,n}$ holds: $\psi(X, Y) = \{\psi((i, j), (i', j')) \mid (i, j) \in X, (i', j') \in Y\}$. We are now able to specify the cardinal direction function *dir* in Definition 5.3.

Definition 5.3. Let $A, B \in region$. Then the *cardinal direction function* *dir* is defined as

$$dir(A, B) = \psi(loc(A, OIM(A, B)), loc(B, OIM(A, B)))$$

We apply this definition to our example in Figure 4. With $loc(A, OIM(A, B)) = \{(2, 2), (3, 2)\}$ and $loc(B, OIM(A, B)) = \{(1, 1), (2, 3)\}$ we obtain

$$\begin{aligned} dir(A, B) &= \psi(\{(2, 2), (3, 2)\}, \{(1, 1), (2, 3)\}) \\ &= \{\psi((2, 2), (1, 1)), \psi((2, 2), (2, 3)), \psi((3, 2), (1, 1)), \psi((3, 2), (2, 3))\} \\ &= \{SE, W, SW\} \end{aligned}$$

Syntactically, the function *dir* yields a set of basic cardinal directions. Definition 5.4 specifies the semantics of the function result ($\otimes : region \times region \rightarrow region$ denotes the geometric intersection operation between two region objects).

Definition 5.4. Let $A, B \in region$, $I_{m,n}^A = loc(A, OIM(A, B))$, and $I_{m,n}^B = loc(B, OIM(A, B))$. Then $dir(A, B) = \{d_1, \dots, d_k\}$ if the following conditions hold:

- (i) $k \in \mathbb{N}, 1 \leq k \leq 9$
- (ii) $\forall 1 \leq i \leq k : d_i \in CD$
- (iii) $\forall 1 \leq l \leq k \exists (i_1, j_1) \in I_{m,n}^A \exists (i_2, j_2) \in I_{m,n}^B \exists r_1, r_2 \in region$
 $\exists t_{i_1, j_1}, t_{i_2, j_2} \in OIGT(A, B) : r_1 = A \otimes t_{i_1, j_1} \wedge r_1 \neq \emptyset \wedge$
 $r_2 = B \otimes t_{i_2, j_2} \wedge r_2 \neq \emptyset \wedge \psi((i_1, j_1), (i_2, j_2)) = d_l$
- (iv) $\forall (i_1, j_1) \in I_{m,n}^A \forall (i_2, j_2) \in I_{m,n}^B \exists r_1, r_2 \in region$
 $\exists t_{i_1, j_1}, t_{i_2, j_2} \in OIGT(A, B) : r_1 = A \otimes t_{i_1, j_1} \wedge r_1 \neq \emptyset \wedge$
 $r_2 = B \otimes t_{i_2, j_2} \wedge r_2 \neq \emptyset \wedge \psi((i_1, j_1), (i_2, j_2)) \in dir(A, B)$

Condition (iii) requires that for all basic cardinal directions d_l of the result at least one region component r_1 of A and at least one region component r_2 of B exist that are located in possibly different tiles of the objects interaction grid, are non-empty, and have the basic cardinal direction d_l towards each other. Condition (iv) ensures that the basic cardinal directions between all region components r_1 of A and r_2 of B are contained in the result set. Using the result from $dir(A, B)$, we can say regarding Figure 4 that “Object A is *partly southeast*, *partly west*, and *partly southwest* of object B ”.

In the same manner, we can also determine the inverse cardinal direction, that is, the cardinal direction between B and A , for the example above:

$$\begin{aligned} dir(B, A) &= \psi(\{(1, 1), (2, 3)\}, \{(2, 2), (3, 2)\}) \\ &= \{\psi((1, 1), (2, 2)), \psi((1, 1), (3, 2)), \psi((2, 3), (2, 2)), \psi((2, 3), (3, 2))\} \\ &= \{NW, E, NE\} \end{aligned}$$

Similarly, we derive from $dir(B, A)$ that “Object B is *partly northwest*, *partly east*, and *partly northeast* of object A ”. We can see that the results of $dir(A, B)$ and $dir(B, A)$ are consistent with the converseness property of cardinal directions. This is an important feature of the objects interaction matrix model.

The following Lemma 5.5 shows that our model satisfies the property of converseness.

LEMMA 5.5. *Let $A, B \in region$. Then $dir(A, B) = inv(dir(B, A))$.*

Proof. Let $dir(B, A) = \{inv(d_1), \dots, inv(d_k)\}$ with $d_l \in CD$ for all $1 \leq l \leq k$. Then there are elements $(i', j') \in I_{m,n}^B$ and $(i, j) \in I_{m,n}^A$ for all $1 \leq l \leq k$ such that $\psi((i', j'), (i, j)) = inv(d_l)$. This means $inv(\psi((i', j'), (i, j))) = inv(inv(d_l)) = d_l$ for all $1 \leq l \leq k$. Using the second property of the interpretation table, we obtain that $\psi((i, j), (i', j')) = d_l$ for all $1 \leq l \leq k$. Since $(i, j) \in I_{m,n}^A$ is the first argument and $(i', j') \in I_{m,n}^B$ is the second argument of ψ , this means that $d_l \in dir(A, B)$ for all $1 \leq l \leq k$. It follows that $dir(A, B) = inv(dir(B, A))$ and that $dir(A, B) = \{d_1, \dots, d_k\}$. \square

6. COMPARISON OF THE OIM MODEL TO PAST APPROACHES

We now review some of the problems raised in the Sections 1 and 2 and show that our OIM model overcomes them. As a case study, we take the scenario of the two countries Argentina and Brazil in Figure 12 and compare the cardinal directions between them as they result from the different cardinal direction models discussed in Section 2.

The first problem is the problem of *object shapes consideration* that leads to misleading and cognitively unexpected results. Models that capture directions between region objects have evolved from reducing these objects to points, to the use of minimum bounding rectangles to approximate their extent, and ultimately to the final goal of considering their shapes. The point approximation models approximate spatial objects into points and determine the cardinal direction relations according to their locations. Table III shows the cardinal directions between Argentina and Brazil on the basis of different cardinal direction models. While a main part of Argentina is located southwest of a main part of Brazil, other parts of Argentina and Brazil are in a different directional relationship to each other. This shows that



Fig. 12. Determining the cardinal direction between Argentina and Brazil

the cardinal directions derived from point approximation models are too coarse to represent the spatial reality. In the group of MBR approximation models, the Directional-Relation Matrix (DRM) model is superior to Papadias's model due to the fact that it captures the shape of the target object. However, it only represents an intermediate step between the MBR model and our final goal of full shape consideration because only the shape of one object is considered and the shape of the other object does not contribute at all. The OIM model that we propose in this paper is the first model that considers the shapes of both region objects.

Unlike in Papadias's model and in the 2D string model in which both region objects play the same role, in the CDR model, the DRM model, Safar's model as well as all point approximation models, a target object is tested for intersection against the tiles created by and around a reference object. As a result, components of the target object inside different tiles contribute to the final cardinal directions while the reference object contributes as a whole object. This treatment causes imprecision. Taking the DRM model as an example, let $dir_{DRM}(A, B)$ be the function that determines the cardinal direction for two simple regions A and B in the DRM model where A is the target object and B is the reference object. Then the cardinal direction between Argentina (A) and Brazil (B) is $dir_{DRM}(A, B) = \{sameLocation, S\}$. This is imprecise because for the major part of Brazil, Argentina lies to the *south-west*, and it also lies to the *west* of some part of Brazil. In our OIM model, both

Model α		$dir_{\alpha}(A, B)$	$dir_{\alpha}(B, A)$
Point approximation	Haar	$\{S\}$	$\{N\}$
	Peuquet	$\{S\}$	$\{N\}$
	Frank	$\{SW\}$	$\{NE\}$
MBR approximation	Papadias	$\{weak_bounded_south\}$	$\{weak_north\}$
	DRM	$\{sL^{\dagger}, S\}$	$\{sL^{\dagger}, NW, N, NE, E\}$
	CDR	$\{sL^{\dagger}, S\}$	$\{sL^{\dagger}, N, E\}$
	2D-string	$\{S\}$	$\{NW, N, NE\}$
	Abdelmoty	$\{sL^{\dagger}, S\}$	$\{sL^{\dagger}, N, E\}$
	OIM	$\{S, W, SW, O, SE\}$	$\{N, E, NE, O, NW\}$
MBC approximation	Safar	$\{sL^{\dagger}, S, W\}$	$\{sL^{\dagger}, N, E\}$

[†] sL means *sameLocation* (corresponds to the *origin* (O) in the OIM model)

Table III. Cardinal directions between Argentina (A) and Brazil (B) in Figure 12, as they are derived from different models

objects are considered, and thus both contribute to the final cardinal direction. Our model yields the result $dir_{OIM}(A, B) = \{SE, S, SW, W, O\}$, which captures the cardinal directions precisely in the sense of Definition 5.4.

The *converseness* problem is a common problem shared by most of the MBR approximation models. It means that the models generate inconsistent results when swapping their operand objects. Table III reveals that Papadias’s model, the DRM model, the CDR model, the 2D-string model, and Abdelmoty’s model do not maintain the converseness property, that is, $dir_{\alpha}(A, B) \neq inv(dir_{\alpha}(B, A))$ for $\alpha \in \{Papadias, DRM, CDR, 2D-string, Abdelmoty\}$. Only the OIM model supports converseness, that is, $dir_{OIM}(A, B) = inv(dir_{OIM}(B, A))$. Hence, by applying the OIM model, we obtain consistent results that correspond to human intuition.

Last but not least, we consider the problem of *complex objects support*. Most of the MBR approximation models have originally been designed for simple regions only. Since all these models are based on the minimum bounding rectangle approximation of at least one object, an extension to complex regions and their minimum bounding rectangles is feasible without difficulty. However, this procedure usually generates rather poor results. For example, in Figure 4a, if we take the minimum bounding rectangle of the entire object B , then object A is to the *weak_bounded_south* of object B according to Papadias’s model, and object A is to the *sameLocation* and *south* of object B according to the DRM model. Both results are imprecise since the western direction of A to one component of B is not captured. Although variants exist for the models to handle complex objects more precisely, considerable efforts are required. Our model natively supports complex objects and is able to yield much more precise results. For the same example in Figure 4a, our model generates the result $dir(A, B) = \{SE, W, SW\}$, which describes object A to be partly *southeast*, partly *west*, and partly *southwest* of object B .

7. DIRECTIONAL PREDICATES FOR QUERYING BASED ON THE OIM MODEL

Using the Objects Interaction Matrix model with its tiling phase and its interpretation phase described in the previous sections, we can identify the cardinal directions between any two given complex region objects. However, cardinal directions are not predicates with the signature $region \times region \rightarrow bool$. To integrate cardinal directions into spatial databases as selection and join conditions in spatial queries, *directional predicates* are needed and have to be formally defined. For example, a query like “Find all states that are strictly north of Florida” requires a directional predicate like *strictly_north_of* as a selection condition of a spatial join. Assuming a relation *states* with attributes *sname* of type *string* and *loc* of type *region*, we can express the query in an SQL-like style as follows:

```
SELECT s1.sname FROM states s1, states s2
WHERE s2.sname = 'Florida' and s1.loc strictly_north_of s2.loc;
```

The *dir* function, which produces the final cardinal directions between any two complex region objects A and B , yields a subset of the set $CD = \{N, NW, W, SW, S, SE, E, NE, O\}$ of basic cardinal directions. As a result, a total number of $2^9 - 1 = 511$ cardinal directions can be identified. Therefore, a maximum of 511 directional predicates can be defined to provide an *exclusive* and *complete* coverage of all possible *directional relationships*. However, we can assume that users will not be inter-

ested in such a large, overwhelming, and predefined collection of detailed predicates since they will find it difficult to distinguish, remember, and handle them. Instead, a reduced and manageable set of predicates will be preferred. Such a set should be user-defined and/or application specific since different applications may have different criteria for the distinction of directional relationships. For example, one application could require a clear distinction between the cardinal directions *north*, *northeast*, and *northwest*, whereas another application could abandon a distinction between the three predicates and regard them all as *northern*. Thus, flexibility is needed to enable users to define their own set of directional predicates and select their own names for these predicates. For this purpose, we provide two *levels* of directional predicates. Level 1 comprises so-called *existential directional predicates*, and Level 2 contains so-called *derived directional predicates*. The predicates of the higher level are derived and built from the predicates of the lower level. All predicates ease the task of formulating complex direction queries.

Level 1 comprises nine predefined *existential directional predicates* that we specify in Definition 7.1 and that ensure the existence of a particular basic cardinal direction between at least one part of a region object A and at least one part of a region object B . These predicates provide an interface for users to define their own derived directional predicates at Level 2. Therefore, we assume that the implementation of the nine existential directional predicates is provided in a spatial type system (spatial algebra, spatial extension package) and that these predicates can be embedded into spatial SQL queries.

Definition 7.1. Let $A, B \in \text{region}$. Then the *existential directional predicates* are defined as

$A \text{ north_of } B$	$\stackrel{\text{def}}{\iff}$	$N \in \text{dir}(A, B)$
$A \text{ west_of } B$	$\stackrel{\text{def}}{\iff}$	$W \in \text{dir}(A, B)$
$A \text{ south_of } B$	$\stackrel{\text{def}}{\iff}$	$S \in \text{dir}(A, B)$
$A \text{ east_of } B$	$\stackrel{\text{def}}{\iff}$	$E \in \text{dir}(A, B)$
$A \text{ origin_with } B$	$\stackrel{\text{def}}{\iff}$	$O \in \text{dir}(A, B)$
$A \text{ northwest_of } B$	$\stackrel{\text{def}}{\iff}$	$NW \in \text{dir}(A, B)$
$A \text{ southwest_of } B$	$\stackrel{\text{def}}{\iff}$	$SW \in \text{dir}(A, B)$
$A \text{ northeast_of } B$	$\stackrel{\text{def}}{\iff}$	$NE \in \text{dir}(A, B)$
$A \text{ southeast_of } B$	$\stackrel{\text{def}}{\iff}$	$SE \in \text{dir}(A, B)$

For example, $A \text{ north_of } B$ returns *true* if a part of A is located to the north of a part of B ; this does not exclude the existence of other basic cardinal directions between other parts of A and B . Hence, we have an existential viewpoint.

Lemma 7.2 shows that by using the existential directional predicates and the logical operators \neg , \vee , and \wedge , we can obtain a complete coverage and mutual distinction of all possible 511 basic and composite cardinal directions from the OIM model based on the set CD . This will enable users to define any set of composite directional predicates for their own needs and applications.

LEMMA 7.2. Let $\langle d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9 \rangle = \langle N, W, S, E, O, NW, SW, NE, SE \rangle$ be the list of basic cardinal directions, and let $\langle p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9 \rangle =$

$\langle north_of, west_of, south_of, east_of, origin_with, northwest_of, southwest_of, northeast_of, southeast_of \rangle$ be the list of existential directional predicates. Let further $A, B \in region$. Then for any basic or composite cardinal direction provided by $dir(A, B)$, the following Boolean expression returns true:

$$\bigwedge_{\substack{1 \leq i \leq 9 \\ d_i \in dir(A, B)}} p_i \wedge \bigwedge_{\substack{1 \leq j \leq 9 \\ d_j \in CD - dir(A, B)}} \neg p_j$$

Proof. The set $dir(A, B)$ contains all basic cardinal directions that hold between parts of A and parts of B . Hence, for each such basic cardinal direction, the pertaining unique existential directional predicate must hold according to Definition 7.1. This is stated in the first operand of the Boolean expression. Since the valid existential directional predicates do not exclude the remaining ones, which correspond to the basic cardinal directions that are in the set CD but not in the set $dir(A, B)$, this exclusion has to be explicitly expressed by negating the remaining existential directional predicates. This is stated by the second operand. \square

For example, if $dir(A, B) = \{W, NW, N\}$, this corresponds to the unique Boolean expression

$$\begin{aligned} & A \text{ west_of } B \wedge A \text{ northwest_of } B \wedge A \text{ north_of } B \wedge \\ & \neg(A \text{ northeast_of } B) \wedge \neg(A \text{ east_of } B) \wedge \neg(A \text{ southeast_of } B) \wedge \\ & \neg(A \text{ south_of } B) \wedge \neg(A \text{ southwest_of } B) \wedge \neg(A \text{ origin_with } B) \end{aligned}$$

Hence, each of the $2^9 - 1$ possible sets that $dir(A, B)$ can produce corresponds to a unique Boolean expression containing the nine either non-negated or negated existential directional predicates. At least one existential directional predicate must be non-negated.

Based on the nine existential predicates from Level 1, users can specify their own *derived directional predicates* at Level 2. Derived directional predicates are composite directional predicates that are built using one or more of the existential directional predicates and other already defined derived directional predicates. As examples, we provide six categories of derived directional predicates, namely *similarly-oriented directional predicates*, *capped directional predicates*, *strict directional predicates*, *strict similarly-oriented directional predicates*, *strict capped directional predicates*, and *surround directional predicates*. These categories are described in the following.

Similarly-oriented directional predicates. The first category comprises derived directional predicates that contain three existential directional predicates with a similar general orientation as well as possibly other basic cardinal directions. Orientation refers to a combination of one primary basic cardinal direction from $\{north, south, west, east\}$ and two “adjacent” secondary basic cardinal directions from $\{northwest, northeast, southwest, southeast\}$. This leads to the four similarly-oriented directional predicates *northern_of*, *southern_of*, *western_of*, and *eastern_of*. Definition 7.3 shows the specification of the predicate *northern*.

Definition 7.3. Let $A, B \in region$. Then the *similarly-oriented directional predicate northern* is defined as:

$$A \text{ northern_of } B \stackrel{\text{def}}{\Leftrightarrow} A \text{ north_of } B \vee A \text{ northwest_of } B \vee A \text{ northeast_of } B$$

The other three similarly-oriented directional predicates are defined in an analogous manner. Note again that other possible basic cardinal directions are not excluded since no statement is made about them. The disjunction of the three existential directional predicates allows several possible cardinal directions in northern direction. This can be expressed by the following equivalent definition of the predicate *northern_of*:

$$A \text{ northern_of } B \stackrel{\text{def}}{\Leftrightarrow} \exists C \in \{\{N\}, \{NW\}, \{NE\}, \{N, NW\}, \{N, NE\}, \{NW, NE\}, \{N, NW, NE\}\} : C \subseteq \text{dir}(A, B)$$

Capped directional predicates. The second category includes the derived directional predicates *north_cap_of*, *south_cap_of*, *west_cap_of*, and *east_cap_of*. They contain a set of “adjacent” existential directional predicates which form the shape of a “cap” such that a region object *A* half surrounds a region object *B*. Definition 7.4 shows the specification of the predicate *north_cap_of*. The other three capped directional predicates are defined in an analogous manner.

Definition 7.4. Let $A, B \in \text{region}$. Then the *capped directional predicate north_cap_of* is defined as:

$$A \text{ north_cap_of } B \stackrel{\text{def}}{\Leftrightarrow} A \text{ west_of } B \wedge A \text{ northwest_of } B \wedge A \text{ north_of } B \wedge A \text{ northeast_of } B \wedge A \text{ east_of } B$$

Strict directional predicates. The third category comprises derived directional predicates that are stricter than the other predicates since two region objects *A* and *B* may only be in one basic cardinal direction to each other. Definition 7.5 shows the specification of the predicate *strictly_north_of*. The other strict directional predicates *strictly_south_of*, *strictly_east_of*, *strictly_west_of*, *strictly_origin_with*, *strictly_northeast_of*, *strictly_northwest_of*, *strictly_southeast_of*, and *strictly_southwest_of* are defined in an analogous manner.

Definition 7.5. Let $A, B \in \text{region}$. Then the *strict directional predicate strictly_north_of* is defined as:

$$\begin{aligned} A \text{ strictly_north_of } B \stackrel{\text{def}}{\Leftrightarrow} & A \text{ north_of } B \wedge \\ & \neg(A \text{ south_of } B) \wedge \neg(A \text{ west_of } B) \wedge \\ & \neg(A \text{ east_of } B) \wedge \neg(A \text{ northwest_of } B) \wedge \\ & \neg(A \text{ northeast_of } B) \wedge \neg(A \text{ southwest_of } B) \wedge \\ & \neg(A \text{ southeast_of } B) \wedge \neg(A \text{ origin_with } B) \end{aligned}$$

Strict similarly-oriented directional predicates. The fourth category comprises derived directional predicates that are the strict versions of the similarly-oriented directional predicates in the first category. This means that two region objects *A* and *B* are arranged in exactly one general orientation. Definition 7.6 shows the specification of the *strict similarly-oriented directional predicate strictly_northern*.

Definition 7.6. Let $A, B \in \text{region}$. Then the *strict similarly-oriented directional predicate* *strictly_northern_of* is defined as:

$$\begin{aligned} A \text{ strictly_northern_of } B &\stackrel{\text{def}}{\iff} A \text{ northern_of } B \wedge \\ &\quad \neg(A \text{ west_of } B) \wedge \neg(A \text{ southwest_of } B) \wedge \\ &\quad \neg(A \text{ south_of } B) \wedge \neg(A \text{ east_of } B) \wedge \\ &\quad \neg(A \text{ southeast_of } B) \wedge \neg(A \text{ origin_with } B) \end{aligned}$$

In other words, if A is strictly northern of B , then we obtain $\text{dir}(A, B) \in \{\{N\}, \{NW\}, \{NE\}, \{N, NW\}, \{N, NE\}, \{NW, NE\}, \{N, NW, NE\}\}$. The remaining directional predicates *strictly_eastern_of*, *strictly_western_of*, and *strictly_southern_of* are defined in an analogous manner.

Strict capped directional predicates. The fifth category comprises derived directional predicates that are the strict versions of the capped directional predicates in the second category. This means that two region objects A and B are in exactly one “capped” orientation with respect to each other. Definition 7.7 shows the specification of the strict capped directional predicate *strictly_north_cap_of*. The other strict capped directional predicates *strictly_east_cap_of*, *strictly_west_cap_of*, and *strictly_south_cap_of* are defined in an analogous manner.

Definition 7.7. Let $A, B \in \text{region}$. Then the *strict capped directional predicate* *strictly_north_cap_of* is defined as:

$$\begin{aligned} A \text{ strictly_north_cap_of } B &\stackrel{\text{def}}{\iff} A \text{ north_cap_of } B \wedge \\ &\quad \neg(A \text{ origin_with } B) \wedge \neg(A \text{ southwest_of } B) \wedge \\ &\quad \neg(A \text{ south_of } B) \wedge \neg(A \text{ southeast_of } B) \end{aligned}$$

Surround Directional Predicates. The sixth and final category of directional predicates contains the single predicate *surrounds*. Its meaning is that two region objects A and B are in multiple cardinal direction relations to each other, with one object totally enveloping the other.

Definition 7.8. Let $A, B \in \text{region}$. Then the directional predicate *surround* is defined as:

$$\begin{aligned} A \text{ surrounds } B &\stackrel{\text{def}}{\iff} A \text{ north_of } B \wedge A \text{ northeast_of } B \wedge A \text{ east_of } B \wedge \\ &\quad A \text{ southeast_of } B \wedge A \text{ south_of } B \wedge A \text{ southwest_of } B \wedge \\ &\quad A \text{ west_of } B \wedge A \text{ northwest_of } B \wedge A \text{ origin_with } B \end{aligned}$$

As we have seen, all derived directional predicates are defined on the basis of the existential directional predicates given in Definition 7.1. We have proposed the derived directional predicates here only as examples. It is up to the users to define their own predicates and to select their own predicate names according to their needs and their applications. But for this purpose, they need data definition language (DDL) support in SQL in order to be able to specify their own directional predicates and then to use them in SQL queries. We propose a new **CREATE DIRPRED** clause that enables users to syntactically specify derived directional predicates in an easy manner. For this, we employ the logical operators $\&$, \mid , and \sim for the logical conjunction, disjunction, and negation of parameterless Boolean functions. Since

all directional predicates are binary predicates, we omit their arguments for reasons of simplicity and better readability. As examples we show the SQL specification of the directional predicates *northern_of*, *strictly_northern_of*, and *strictly_surrounds*. The latter predicate has not been defined before and is the strict version of the predicate *surrounds* with the difference that the first operand region may not be located within the origin.

```
CREATE DIRPRED northern_of
(north_of | northwest_of | northeast_of)

CREATE DIRPRED strictly_northern_of
(northern_of & ~west_of & ~southwest_of & ~south_of &
~east_of & ~southeast_of & ~origin_with)

CREATE DIRPRED strictly_surrounds
(north_of & northeast_of & east_of & southeast_of & south_of &
southwest_of & west_of & northwest_of & ~origin_with)
```

We have not used the logical operators **AND**, **OR**, and **NOT** provided by SQL for the specification of **CREATE DIRPRED** clauses since they apply to parameterized Boolean expressions. We show the relationship between the logical operators **&**, **|**, and **~** and the logical SQL operators **AND**, **OR**, and **NOT** for the first clause with $A, B \in \text{region}$:

$$(north_of \mid northwest_of \mid northeast_of)(A, B) \Leftrightarrow (north_of(A, B) \text{ OR } northwest_of(A, B) \text{ OR } northeast_of(A, B))$$

The use of the directional predicates in SQL queries is quite standard and therefore familiar to the user. However, the semantics that can be expressed by these predicates can be quite different and powerful. We show this expressiveness in a scenario of the states and national parks in the U.S. and use the relation tables

```
states(sname:string, area:region)
national_parks(pname:string, area:region)
```

The schemas make use of the spatial data type *region*. This scenario also demonstrates that our OIM model cannot only be based on a Cartesian coordinate system but also on a geographic coordinate system representing coordinates by means of latitudes and longitudes. In our scenario, we consider the latitude lines and longitude lines along the earth surface as the reference horizontal and vertical lines to compute the partitioning lines for the objects interaction grid. We pose the following queries:

Q1: Determine the states that are western of the state of Wyoming.

```
SELECT S2.sname FROM states S1, states S2
WHERE S1.sname = 'Wyoming' AND S2.area western_of S1.area
```

Q2: Determine the states that are located exclusively western of the state of Wyoming.

```
SELECT S2.sname FROM states S1, states S2
WHERE S1.sname = 'Wyoming' AND S2.area strictly_western_of S1.area
```

Q3: Determine the states that are both western of and southern of the state of Wyoming.

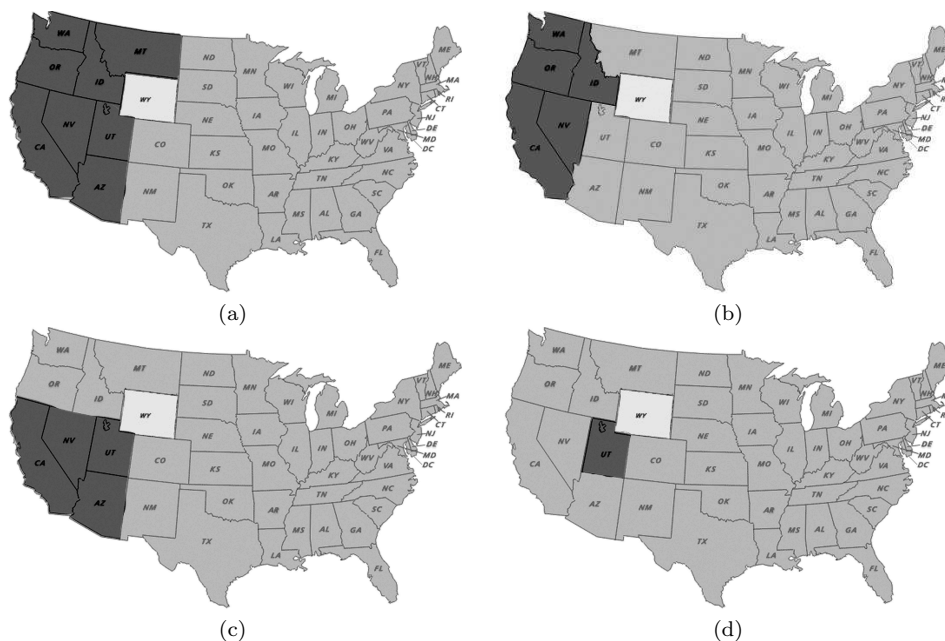


Fig. 13. Results for the queries Q1 (a), Q2 (b), Q3 (c), and Q4 (d).

```

SELECT S2.sname FROM states S1, states S2
WHERE S1.sname = 'Wyoming' AND
      S2.area western_of S1.area AND S2.area southern_of S1.area

```

Q4: Determine the states that are partly to the west of and partly to the south of the state of Wyoming.

```

SELECT S2.sname FROM states S1, states S2
WHERE S1.sname = 'Wyoming' AND
      S2.area west_of S1.area AND S2.area south_of S1.area

```

Q5: Determine the national parks that are at least northern or eastern of Tennessee.

```

SELECT P.pname FROM national_parks P, states S
WHERE S.sname = 'Tennessee' AND
      (P.area northern_of S.area OR P.area eastern_of S.area)

```

Q6: Determine the national parks that are only northern or eastern of Tennessee.

```

SELECT P.pname FROM national_parks P, states S
WHERE S.sname = 'Tennessee' AND
      (P.area strictly_northern_of S.area OR
       P.area strictly_eastern_of S.area)

```

Q7: Determine the national parks that surround Tennessee.

```

SELECT P.pname FROM national_parks P, states S
WHERE S.sname = 'Tennessee' AND P.area surrounds S.area

```

The first four queries are illustrated in Figure 13. Q1 returns the eight states WA, MT, OR, ID, CA, NV, UT, and AZ (Figure 13a). Due to the existential

character of the *western_of* predicate, a state can also be in another directional relationship to Wyoming. For example, *MT* is also north of *WY*. Q2 has the more restricted query condition *strictly_western_of*; thus, the three states *MT*, *UT*, and *AZ* are excluded because they have parts that are either north of state *WY*, south of state *WY*, or together with a part of state *WY* in the origin *O* (Figure 13b). The state *ID*, however, shares the border with *WY* that is defined by the longitude line between them. Hence, the directional relationship *strictly_western_of* holds for them. The result of Q3 shows that only *UT*, *AZ*, *CA*, and *NV* are to the western and to the southern of *WY* (Figure 13c). Q4 finds out that only *UT* is to the west and to the south of *WY* (Figure 13d). Q5 requires the existence of any northern or southern direction so that a national park is located in that direction to Tennessee. This query admits other directional relationships too. Q6 looks exclusively for a northern or eastern direction of a national park to Tennessee. Q7 asks for surrounding national parks. It is possible that a national park and Tennessee share the origin *O*. The use of the predicate *strictly_surrounds* would prevent this.

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have laid the foundation of a novel concept, called Objects Interaction Matrix (OIM), for determining cardinal directions between complex region objects. After a discussion of the available cardinal direction models and a determination of their weaknesses, we have introduced and formally defined our OIM model that consists of a tiling phase and an interpretation phase and solves all the identified problems of previous approaches. We have shown how different kinds of directional predicates can be derived from this model and how these predicates can be employed in spatial queries in an SQL-like query language.

In the future, we plan to extend our approach to handle two complex point objects, two complex line objects, and all mixed combinations of spatial data types. As long as complex point objects and complex line objects as operand objects have a two-dimensional extent and thus non-degenerate minimum bounding rectangles, the concepts presented in this article can be applied to them. However, special cases like horizontal and vertical line objects as well as complex point objects whose points are collinear on a horizontal or vertical line cannot be handled by the OIM model. A further important research issue refers to the design of efficient algorithms for the directional predicates based on the OIM model.

REFERENCES

- ABDELMOTY, A. I. AND WILLIAMS, M. H. 1994. Approaches to the Representation of Qualitative Spatial Relationships for Geographic Databases. In *Int. Workshop on Advanced Geographic Data Modeling*. 204–216.
- ALLEN, J. F. 1983. Maintaining Knowledge about Temporal Intervals. *Journal of the Association for Computing Machinery* 26, 11, 832–843.
- CHANG, S.-K. 1989. *Principles of Pictorial Information Systems Design*. Prentice-Hall, Inc.
- CHANG, S.-K. AND LI, Y. 1988. Representation of Multi-resolution Symbolic/Binary Pictures using 2DH Strings. In *IEEE Workshop on Languages for Automation*. 190–195.
- CHEN, T., SCHNEIDER, M., VISWANATHAN, G., AND YUAN, W. 2010. The Objects Interaction Matrix for Modeling Cardinal Directions in Spatial Databases. In *Int. Conf. on Database Systems for Advanced Applications*. 218–232.

- CLEMENTINI, E. AND BILLEN, R. 2006. Modeling and Computing Ternary Projective Relations between Regions. *IEEE Trans. on Knowledge and Data Engineering* 18, 6, 799–814.
- CLEMENTINI, E. AND DI FELICE, P. 1996. A Model for Representing Topological Relationships between Complex Geometric Features in Spatial Databases. *Information Sciences* 90, 1-4, 121–136.
- EGENHOFER, M. J. 1994. Spatial SQL: A Query and Presentation Language. *IEEE Trans. on Knowledge and Data Engineering* 6, 1, 86–94.
- FRANK, A. U. 1991. Qualitative Spatial Reasoning about Cardinal Directions. In *Int. Research Symp. on Computer-based Cartography and GIScience – AutoCarto*. Vol. 6. 148–167.
- FRANK, A. U. 1992. Qualitative Spatial Reasoning about Distances and Directions in Geographic Space. *Journal of Visual Languages and Computing* 3, 343–371.
- FRANK, A. U. 1996. Qualitative Spatial Reasoning: Cardinal Directions as an Example. *Int. Journal of Geographical Information Systems* 10, 3, 269–290.
- GOYAL, R. AND EGENHOFER, M. 2000a. Cardinal Directions between Extended Spatial Objects. Unpublished manuscript.
- GOYAL, R. AND EGENHOFER, M. J. 1997. The Direction-Relation Matrix: A Representation for Directions Relations between Extended Spatial Objects. In *The Annual Assembly and the Summer Retreat of the University Consortium for Geographic Information Systems Science*.
- GOYAL, R. K. AND EGENHOFER, M. J. 2000b. Consistent Queries over Cardinal Directions Across Different Levels of Detail. In *11th Int. Conf. on Database and Expert Systems Applications*. 876–880.
- GÜTING, R. H. 1988. Geo-Relational Algebra: A Model and Query Language for Geometric Database Systems. In *Int. Conf. on Extending Database Technology*. 506–527.
- HAAR, R. 1976. Computational Models of Spatial Relations. Tech. Rep. TR-478, Computer Science, University of Maryland, College Park, MD, USA.
- ISLI, A. AND COHN, A. G. 2000. A New Approach to Cyclic Ordering of 2D Orientations Using Ternary Relation Algebras. *Artificial Intelligence* 122, 1-2, 137–187.
- JING, Z., MARK, D. M., AND ZHENG-RONG, Z. 2005. The New Reference Frame about the Spatial Orientation Expression for Way-finding. In *Int. Symp. on Spatial-temporal Modeling, Spatial Reasoning, Spatial Analysis, Data Mining, Data Fusion*.
- KIM, B. AND UM, K. 1999. 2D+ String: A Spatial Metadata to Reason Topological and Directional Relationships. In *Int. Conf. on Scientific and Statistical Database Management*. 112–121.
- NABIL, M., SHEPHERD, J., AND NGU, A. H. H. 1995. 2D Projection Interval Relationships: A Symbolic Representation of Spatial Relationships. In *Int. Symp. on Advances in Spatial Databases*. 292–309.
- PAPADIAS, D., EGENHOFER, M., AND SHARMA, J. 1996. Hierarchical Reasoning about Direction Relations. In *4th ACM Symp. on Advances in Geographic Information Systems*. 105–112.
- PAPADIAS, D. AND EGENHOFER, M. J. 1997. Algorithms for Hierarchical Spatial Reasoning. *GeoInformatica* 1, 3, 251–273.
- PAPADIAS, D. AND SELLIS, T. 1994. Qualitative Representation of Spatial Knowledge in Two-dimensional Space. *The VLDB Journal* 3, 4, 479–516.
- PAPADIAS, D. AND THEODORIDIS, Y. 1997. Spatial Relations, Minimum Bounding Rectangles, and Spatial Data Structures. *Int. Journal of Geographical Information Science* 11, 2, 111–138.
- PEUQUET, D. J. AND ZHANG, C.-X. 1987. An Algorithm to Determine the Directional Relationship between Arbitrarily-shaped Polygons in the Plane. *Pattern Recognition* 20, 1, 65–74.
- SAFAR, M. AND SHAHABI, C. 1999. 2D Topological and Direction Relations in the World of Minimum Bounding Circles. In *Int. Database Engineering and Applications Symposium*. 239–247.
- SCHNEIDER, M. 1997. *Spatial Data Types for Database Systems - Finite Resolution Geometry for Geographic Information Systems*. Vol. LNCS 1288. Springer-Verlag.
- SCHNEIDER, M. AND BEHR, T. 2006. Topological Relationships between Complex Spatial Objects. *ACM Trans. on Database Systems* 31, 1, 39–81.

- SHEKHAR, S., LIU, X., AND CHAWLA, A. 1999. An Object Model of Direction and Its Implications. *GeoInformatica* 3, 4, 357–379.
- SKIADOPOULOS, S. AND KOUBARAKIS, M. 2004. Composing Cardinal Direction Relations. *Artificial Intelligence* 152, 2, 143–171.
- SKIADOPOULOS, S., SARKAS, N., SELLIS, T., AND KOUBARAKIS, M. 2007. A Family of Directional Relation Models for Extended Objects. *IEEE Trans. on Knowledge and Data Engineering* 19, 8, 1116–1130.
- WORBOYS, M. AND BOFAKOS, P. 1993. A Canonical Model for a Class of Areal Spatial Objects. In *3rd Int. Symp. on Advances in Spatial Databases*. LNCS 692. Springer-Verlag, 36–52.
- ZHU, H., SU, J., AND IBARRA, O. H. 2001. On Multi-way spatial Joins with Direction Predicates. In *7th Int. Symp. on Advances in Spatial and Temporal Databases*. 217–235.

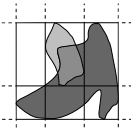
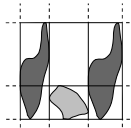
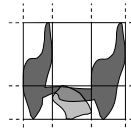
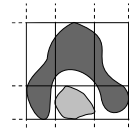
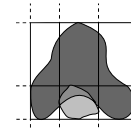
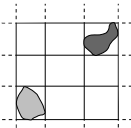
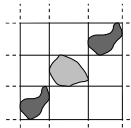
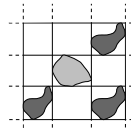
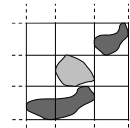
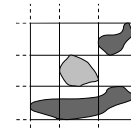
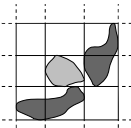
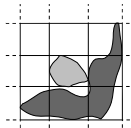
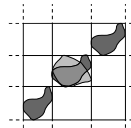
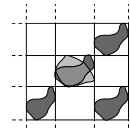
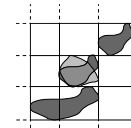
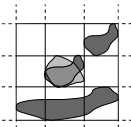
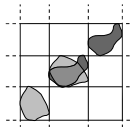
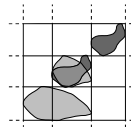
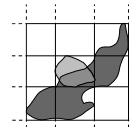
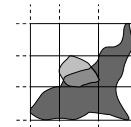
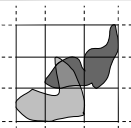
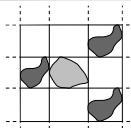
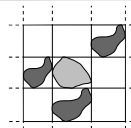
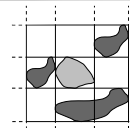
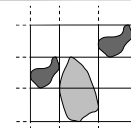
Received September 2011

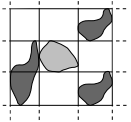
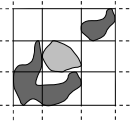
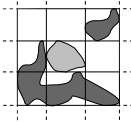
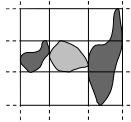
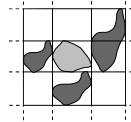
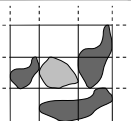
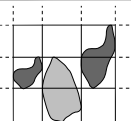
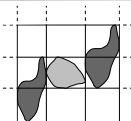
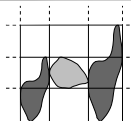
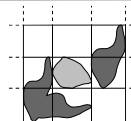
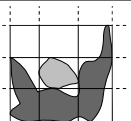
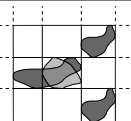
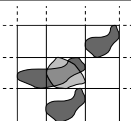
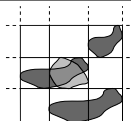
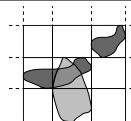
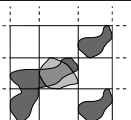
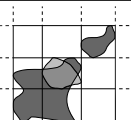
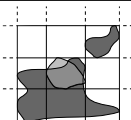
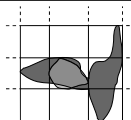
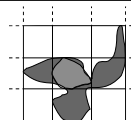
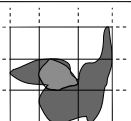
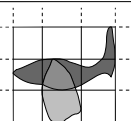
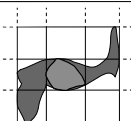
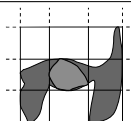
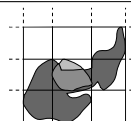
Appendix: Representative OIM Matrices and Prototypical Spatial Configurations

In this appendix, we show the representative objects interaction matrices and prototypical spatial configurations with their objects interaction grids for all 165 matrix groups representing 1677 objects interaction matrices. For each representative matrix, we provide its matrix number (in *italics*), the identifier of the group to which the matrix belongs (in **bold**), the total number of matrices that belong to this matrix group (in parentheses), and whether the matrix is only valid for complex regions (*c*) or also for simple regions (*s*).

1 (1) <i>s 1</i>	2 (4) <i>s 2</i>	3 (8) <i>s 3</i>	4 (4) <i>s 8</i>	5 (4) <i>c 9</i>
$\begin{pmatrix} 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 3 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 1 & 2 & 1 \end{pmatrix}$
6 (4) <i>s 10</i>	7 (4) <i>s 11</i>	8 (8) <i>c 22</i>	9 (8) <i>s 23</i>	10 (4) <i>s 24</i>
$\begin{pmatrix} 1 & 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 3 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 1 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}$
11 (16) <i>s 25</i>	12 (8) <i>s 27</i>	13 (8) <i>c 28</i>	14 (16) <i>s 29</i>	15 (8) <i>s 52</i>
$\begin{pmatrix} 0 & 1 \\ 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 \\ 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$
16 (8) <i>s 53</i>	17 (16) <i>c 106</i>	18 (16) <i>c 107</i>	19 (16) <i>c 108</i>	20 (16) <i>s 109</i>
$\begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \end{pmatrix}$

21(8) s 110	22(16) s 111	23(16) c 112	24(16) s 113	25(8) c 122
$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$
26(16) c 123	27(8) s 124	28(16) s 125	29(8) c 126	30(8) c 127
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 1 & 2 \end{pmatrix}$
31(8) s 129	32(8) s 131	33(16) c 132	34(16) c 133	35(16) s 134
$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 3 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 3 & 0 \end{pmatrix}$
36(16) s 135	37(8) s 136	38(16) s 137	39(16) s 138	40(16) s 139
$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 3 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 2 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 2 & 2 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 2 & 3 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 2 & 3 & 1 \end{pmatrix}$
41(16) c 141	42(16) s 143	43(8) c 166	44(8) s 167	45(16) c 171
$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 3 & 1 \\ 1 & 0 & 1 \end{pmatrix}$

46(16) s 173 $\begin{pmatrix} 0 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ 	47(8) c 195 $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \end{pmatrix}$ 	48(8) s 197 $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 3 & 1 \end{pmatrix}$ 	49(8) s 213 $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \end{pmatrix}$ 	50(8) s 215 $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \end{pmatrix}$ 
51(4) s 546 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix}$ 	52(4) c 547 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ 	53(8) c 548 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ 	54(16) c 549 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ 	55(16) c 550 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ 
56(8) c 553 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ 	57(8) s 554 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ 	58(4) c 555 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ 	59(8) c 556 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ 	60(16) c 557 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ 
61(16) c 558 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ 	62(4) c 559 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 2 & 0 & 0 \end{pmatrix}$ 	63(16) c 560 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 2 & 2 & 0 \end{pmatrix}$ 	64(8) s 563 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ 	65(8) s 564 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ 
66(8) s 566 $\begin{pmatrix} 0 & 0 & 1 \\ 0 & 3 & 1 \\ 2 & 2 & 0 \end{pmatrix}$ 	67(8) c 567 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 	68(8) c 568 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ 	69(16) c 569 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ 	70(16) c 570 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 2 & 0 \end{pmatrix}$ 

71(16) c 572 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ 	72(8) c 573 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ 	73(16) c 574 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ 	74(8) c 575 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ 	75(16) c 576 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ 
76(16) c 577 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ 	77(16) c 578 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix}$ 	78(8) c 579 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ 	79(16) c 580 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ 	80(16) c 581 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ 
81(16) s 582 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$ 	82(8) c 583 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ 	83(8) c 584 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 0 & 1 & 0 \end{pmatrix}$ 	84(16) c 585 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 0 & 1 & 1 \end{pmatrix}$ 	85(16) c 586 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 0 & 2 & 0 \end{pmatrix}$ 
86(16) c 588 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}$ 	87(8) c 589 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 1 & 1 & 0 \end{pmatrix}$ 	88(16) c 590 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ 	89(8) s 591 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix}$ 	90(16) s 592 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ 
91(16) s 593 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ 	92(16) s 594 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 0 & 2 & 0 \end{pmatrix}$ 	93(8) s 595 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 0 & 0 \end{pmatrix}$ 	94(16) s 596 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ 	95(16) s 597 $\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ 

96(16) s 598	97(8) c 599	98(8) s 600	99(8) s 601	100(8) c 602
$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 0 & 1 \\ 2 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 1 & 0 \\ 0 & 2 & 0 \end{pmatrix}$
101(8) c 603	102(16) c 604	103(16) s 605	104(16) c 607	105(16) c 608
$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 1 & 0 \\ 2 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 1 & 1 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 2 & 0 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$
106(16) s 609	107(8) c 610	108(16) c 612	109(8) c 613	110(8) c 615
$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 0 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 0 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 0 \\ 2 & 2 & 0 \end{pmatrix}$
111(8) s 616	112(16) s 617	113(16) s 618	114(16) s 619	115(8) s 620
$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \\ 2 & 0 & 0 \end{pmatrix}$
116(16) s 621	117(4) c 730	118(8) c 731	119(8) c 732	120(8) s 733
$\begin{pmatrix} 0 & 0 & 1 \\ 2 & 3 & 1 \\ 2 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 2 \\ 0 & 2 & 2 \end{pmatrix}$

121(2) c 741	122(8) c 742	123(8) c 743	124(8) c 745	125(8) c 747
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
126(8) c 751	127(16) c 753	128(2) s 761	129(8) s 762	130(8) s 763
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 2 \\ 0 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 2 & 2 \\ 1 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 2 & 0 \end{pmatrix}$
131(8) s 765	132(8) s 767	133(4) s 770	134(8) s 771	135(16) s 773
$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 2 \\ 0 & 2 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 2 \\ 0 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 2 \\ 1 & 1 & 0 \end{pmatrix}$
136(2) c 776	137(4) c 784	138(8) c 788	139(8) s 790	140(2) s 813
$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 0 & 2 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 1 & 2 \\ 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 1 & 2 \\ 2 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 1 & 2 \\ 2 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 3 & 2 \\ 0 & 1 & 0 \end{pmatrix}$
141(8) s 817	142(8) s 819	143(8) c 848	144(16) c 850	145(4) c 857
$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 3 & 2 \\ 2 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 \\ 2 & 3 & 2 \\ 2 & 2 & 2 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 0 \end{pmatrix}$

146(8) s 858 $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	147(8) c 864 $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	148(16) s 866 $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 3 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	149(4) s 873 $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 0 \end{pmatrix}$	150(8) s 874 $\begin{pmatrix} 0 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
151(4) s 876 $\begin{pmatrix} 0 & 1 & 1 \\ 2 & 0 & 1 \\ 2 & 2 & 0 \end{pmatrix}$	152(8) s 880 $\begin{pmatrix} 0 & 1 & 1 \\ 2 & 1 & 1 \\ 2 & 2 & 0 \end{pmatrix}$	153(4) s 901 $\begin{pmatrix} 0 & 1 & 1 \\ 2 & 3 & 1 \\ 2 & 2 & 0 \end{pmatrix}$	154(2) c 1187 $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	155(8) c 1189 $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 1 & 1 \end{pmatrix}$
156(8) c 1197 $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	157(2) c 1203 $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 1 \end{pmatrix}$	158(8) c 1205 $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	159(8) c 1213 $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	160(4) c 1227 $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 0 & 1 \end{pmatrix}$
161(8) s 1229 $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	162(4) s 1243 $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	163(8) s 1245 $\begin{pmatrix} 1 & 0 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	164(2) s 1375 $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	165(2) s 1391 $\begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}$