Evolution of Region Connection Calculus to VRCC-3D+

Chaman L. Sabharwal* and Jennifer L. Leopold†

Computer Science Department
Missouri University of S&T
Rolla, MO 65409, USA
*chaman@mst.edu
†leopoldj@mst.edu

Qualitative spatial reasoning (QSR) is useful for deriving logical inferences when quantitative spatial information is not available. QSR theories have applications in areas such as geographic information systems, spatial databases, robotics, and cognitive sciences. The existing QSR theories have been applied primarily to 2D. The ability to perform QSR over a collection of 3D objects is desirable in many problem domains. Here we present the evolution (VRCC-3D+) of RCC-based QSR from 2D to both 3D (including occlusion support) and 4D (a temporal component). It is time consuming to construct large composition tables manually. We give a divide-and-conquer algorithm to construct a comprehensive composition table from smaller constituent tables (which can be easily handcrafted). In addition to the logical consistency entailment checking that is required for such a system, clearly there is a need for a spatio-temporal component to account for spatial movements and path consistency (i.e. to consider only smooth transitions in spatial movements over time). Visually, these smooth movement phenomena are represented as a conceptual neighborhood graph. We believe that the methods presented herein to detect consistency, refine uncertainty, and enhance reasoning about 3D objects will provide useful guidelines for other studies in automated spatial reasoning.

Keywords: Qualitative spatial reasoning; region connection calculus; composition; smooth transition; constraint logic programming.

1. Introduction

Qualitative spatial reasoning (QSR) is useful for deriving logical inferences when quantitative spatial information is not available. QSR theories have applications in areas such as geographic information systems (GIS), spatial databases, image processing, robotics, biomedicine, and cognitive sciences. Qualitative reasoning\(^1\) uses qualitative rather than quantitative information, and concerns the issues related to automating human spatial analytical processes. Several region connection calculi have been proposed for use in QSR. Perhaps the most well known model is RCC-8,\(^2\) which uses an axiomatic framework or the 9-intersection model\(^3\) to analyze the spatial relation between two regions. Both RCC-8 and 9-intersection matrix frameworks were developed independently. The calculation of the 9-intersection matrix is particularly computation-intensive for 3D. However, the 9-intersection calculations
actually can be replaced with 2-intersection, 3-intersection, and (at most) 4-intersection calculations for each RCC-8 relation.

For our purposes, the most important aspects of space are shape, time, and occlusion. Mereotopology typically is the basis for QSR when knowledge is vague in terms of quantitative specification, or when quantitative reasoning is intractable. The qualities of spatial 2D objects that are taken into consideration for such analyses include: topology (i.e. connectedness), mereology (i.e. parthood), morphology, dimensionality, measurement, and distance. In spatio-temporal reasoning, time (1D) accounts for the temporal component for occlusion. The existing QSR theories primarily have been applied to 2D data. It is important to make inferences from simple constraint logical statements in 2D. However, as with any knowledge representation and reasoning system, having additional dimensions of information often facilitates more accurate and informative analyses. Yet, research in and implementations of qualitative spatial reasoning mostly have been limited to 2D, even when higher dimensions of data are available. In part, this is due to human cognitive limitations and the increased computational complexity that results when additional information is considered. However, given the enormous amount of 3D image data that continue to become available, as well as the increasing sophistication of mechanical and software systems that require spatial awareness, it is important that efforts to produce 3D spatial reasoning systems continue to progress.

Regardless of the number of dimensions that are considered, QSR is effectively a constraint satisfaction problem. One starts with a finite set of jointly exhaustive and pairwise disjoint (JEPD) relations (i.e. there is exactly one relation that holds between any two objects). Those basic relations can be combined disjunctively to form more complex relations. Composition of basic or complex relations is central to QSR. A composition table is used to derive inferences of conjunctive constraints over pairs of objects. The relation between any two objects can be inferred from the knowledge of connection of the objects with a third object. For instance, suppose that we do not know the relation between objects A and B. We can consult the composition table to narrow down the possibilities for the relation between A and B if we know, for some other object C, the relation between A and C, and the relation between B and C. Presumably, the composition table is pre-computed and stored for all future references.

The composition table also can be used for consistency checking. Its need originates from three types of problems. When data are collected from various sources, of variable quality for spatial databases and GIS, the data can be: (i) inaccurate (e.g. the information about pairs of objects may be inconsistent or impossible), (ii) incomplete (e.g. the information about pairs of objects may be missing; this happens when resolution is to be made on \( n \) parameters, but \( n - 1 \) parameters are known, and we need to find the value of the \( n \)th parameter to satisfy the disjunctive form), or (iii) indefinite (e.g. the information about pairs of objects may be ambiguous; uncertainty occurs when available information is in disjunctive form and additional parameters
are needed to ascertain disjunctive elements). Previously developed algorithms\(^8\) to construct composition tables have required manual or, at best, semi-automatic analysis. Later, in Sec. 4, we present a divide-and-conquer algorithm\(^9\) that can be used to construct a large table automatically.

There are several approaches\(^10\) to represent and reason about spatio-temporal configurations. However, they all are truly 2D. There have been some incomplete attempts to reason about 3D configurations, including LOS\(^11-14\) and ROC\(^6-20\). We may call those 2 1/2 D, for they are not completely three-dimensional.

In contrast, we introduced a 3D reasoning system,\(^12\) RCC-3D (and VRCC-3D), that was based on generalized 2D region connection calculus (GRCC) and supported occlusion in 3D. We later enhanced that model to a more expressive version,\(^13\) VRCC-3D+. Similar to Ref. 2, we started with RCC-8 (in 2D) to construct a spatial calculus in 3D, and extended our calculus to support occlusion. The relations may be conceptually aggregated to permit different levels of decomposition. This work represents substantial advances over previous work in QSR, specifically in terms of: methods to construct comprehensive composition tables (as previously mentioned), identification of smooth transitions to refine uncertainty, pruning of inconsistency, and enhanced reasoning about objects in 3D including occlusion support.

This paper is organized as Sec. 2: Mathematical and RCC related definitions; Sec. 3: Efficient implementation of intersection model for regions using 9-intersection versus 4-intersection; Sec. 4: Composition table, divide-and-conquer algorithm; Sec. 5: Conceptual neighborhoods; Sec. 6: Smooth transition model and algorithms using 4-Intersections; Sec. 7: Path consistency; Sec. 8: Spatio-temporal considerations; Sec. 9: Future directions; Sec. 10: Conclusions.

2. Background

2.1. Mathematical preliminaries

\(R^3\) denotes the three-dimensional space endowed with a distance metric. Here the mathematical notions of subset, proper subset, equal sets, empty set (\(\emptyset\)), universal complement, union, intersection, and relative complement are the same as those typically defined in set theory. The notions of neighborhood, open set, closed set, limit point, boundary, interior, exterior, and closure of sets are as in point-set topology. The interior, boundary, and exterior of any object are disjoint, and their union is the universe.

A set is connected if it cannot be represented as the union of disjoint open sets. For any non-empty bounded set \(A\), we use symbols \(A^c\), \(A^i\), \(A^b\), and \(A^e\) to represent the universal complement, interior, boundary, and exterior of a set \(A\), respectively. \(A\) is weakly connected to \(B\), denoted by \(\mathcal{C}(A, B)\), if \(\bar{A} \cap \bar{B} \neq \emptyset\); that is, \(A^i \cup A^e\) is weakly connected. This is different from the mathematical definition of connectedness where \(A^i \cup A^e\) is disconnected in topology. Another concept is the use of weak equality. Two regions \(A\) and \(B\) are equal if \(A^i = B^i\), \(A^b = B^b\), and \(A^e = B^e\) are true. For our discussion, we assume that every region \(A\) is a non-empty, bounded, regular
closed, connected set without holes; specifically, \( A^b \) is a closed curve in 2D, and a closed surface in 3D.

If \( A \) is a subset of \( B \), it is denoted by \( P(A, B) \) (i.e. \( A \) is part of \( B \)); specifically, this requires that \( A \subseteq B \), or \( A \cap B^c = \emptyset \). Note that, if we allow \( \emptyset \) to be a region, contradictions can arise: \( \emptyset \) is contained in every region, and is disconnected from it.

Mathematically, a set theoretic relation \( R \) from a set \( A \) to a set \( B \) is a subset of the direct product of \( A \) and \( B \), e.g. \( R \subseteq A \times B \). Thus, a binary relation \( R \) on a set \( D \) is a subset of \( D^2: R \subseteq D \times D = D^2 \). With this definition, \( \emptyset \) is a relation. However, it is important to note that \( \emptyset \) is not a spatial relation. It is purely an abstract mathematical relation.

Note that the statement \( A = \emptyset \) means the value of the set \( A \) is empty set, whereas \( A = \neg \emptyset \) means the value of the set \( A \) is some non-empty set. Sometimes the actual value of a non-empty set is immaterial, in which case a qualitative value \( \neg \emptyset \) is used for the set. For predicates, the value is true or false. For example, in a true-false statement, the Boolean expression \( A = \emptyset \) (or \( A = \neg \emptyset \)) is true or false depending on the value of a set \( A \).

Let \( U \) be the universal of objects, and let \( D \) be a finite subset of \( U \). A binary relation \( R \) on \( D \) is a subset of \( D^2 \) and is also referred to as a constraint on \( D \). The constraint instantiation \( R(A, B) \) is a logical value. The constraint \( R \) is satisfied if the logical value is true for some \( (A, B) \in D^2 \). A constraint satisfaction problem is a set of variables, \( V \), and a set \( R \) of relations, such that for each relation \( R \in R \), there is a solution in \( V \). The term relation and constraint are frequently used.

A binary relation \( R \) may have one or more of the following properties:

- Reflexive: \( \forall A \in D \, R(A, A) \)
- Symmetric (self-converse): \( \forall A, B \in D \,(R(A, B) \rightarrow R(B, A)) \)
- Transitive: \( \forall A, B, C \in D \,(R(A, B) \land R(B, C) \rightarrow R(A, C)) \)

Let the set of base relations be denoted by \( B = \{R^1, R^2, \ldots, R^n\} \). The terms JEPD are formally defined here for relations and are used in the rest of the paper. Jointly exhaustive:

\[
\forall A, B \in D \forall R \in B \, R(A, B)
\]

Pairwise disjoint:

\[
\forall A, B \in D \land R_1, R_2 \in B, R_1 \neq R_2 \, R_1(A, B) \rightarrow \neg R_2(A, B)
\]

Here, the set notation is overloaded: if \( P \) and \( Q \) are relations, \( \{P, Q\}(A, B) \) means disjunction \( P(A, B) \lor Q(A, B) \). Therefore, the set \( \{P, Q\} \) is a general relation representing disjunction \( P \lor Q \). In particular, \( \{R^1, R^2, \ldots, R^n\} \) is a general relation; mathematically it looks like a set, but in terms of relations it is disjunctive relation. The constraint network consists of expressions or constraints of the form

\[
\{R^1, R^2, \ldots, R^n\}(A, B) \text{ or } B(A, B).
\]
Composition: \( \forall A, B, C \in \mathcal{D}, \)
\[
R_1(A, B) \land R_2(B, C) \rightarrow R_1^1(A, C) \lor R_2^2(A, C) \lor \ldots \lor R_n^n(A, C)
\]
for a minimal subset \( \mathcal{M} \) of \( B. \)

Or \( \{R_1, R_2, \ldots, R_n\}(A, C) \), or simply \( \mathcal{M}(A, C). \)

Thus, the composition of \( R_1 \) and \( R_2 \) is denoted by \( R_1 \circ R_2 \), which is a complex relation, or a constraint, interchangeably written as disjunction
\[
R_1^1 \lor R_2^2 \lor \ldots \lor R_n^n \text{ or disjunction } \{R_1, R_2, \ldots, R_n\}.
\]

Relation Algebras were introduced by Tarski and applied by Ladkin and Maddox. A relation algebra is a set of relations and operations on relations. It is closed with respect to these operations. In our case for spatial relations, they are the disjunction \( (\lor) \), negation \( (\neg) \), composition \( (\circ) \), and conversion \( (c) \) operations.

2.2. Region connection calculi

In this section, we give a formal (set theoretic) definition for each RCC-8 relation. The fundamental axioms of the RCC-8 model are Parthood and Connectivity. Parthood is a relation that exists between two objects whenever one is a subpart of the other. Mereology, which is the study of Parthood, is the foundation for many QSR theories. Parthood is sufficient for defining spatial relations such as overlap and disjointness. However, Parthood is not adequate to define spatial relations that need to take connectedness into consideration. Topology is the study of Connectivity and Continuous deformations, and, unlike mereology, is based upon the mathematical concepts of points, lines, regions, and objects. Because mereology alone is not sufficient for doing a broad spectrum of qualitative spatial reasoning, the two fields are often combined to form mereotopology, which defines each spatial relation in terms of Parthood and/or Connectivity. For a detailed study of mereotopology, the reader may consult a basic textbook. An extensive survey of work on qualitative spatial representation and reasoning is based on various mereotopological theories. They further suggest a logical classification of QSR theories based on Connectivity, Parthood, and other necessary axioms.

The predicate \( \mathcal{P}(A, B) \) is used to represent “\( A \) is part of \( B \)” if \( A \) is a subset of \( B \); specifically, this is true if \( A \cap B^c = \emptyset \), where \( \emptyset \) is the empty set. The relation \( \mathcal{P} \) is reflexive: \( \forall A \in D(\mathcal{P}(A, A)) \), asymmetric and transitive: \( \forall A, B, C \in D(\mathcal{P}(A, B) \land \mathcal{P}(B, C) \rightarrow \mathcal{P}(A, C)) \). The predicate \( \mathcal{C}(A, B) \) is used to represent “\( A \) is connected to \( B \).” The relation \( \mathcal{C} \) is reflexive: \( \forall A \in D(\mathcal{C}(A, A)) \), symmetric: \( \forall A, B \in D(\mathcal{C}(A, B) \rightarrow \mathcal{C}(B, A)) \), and non-transitive. Note that, parthood can be derived from connectivity: \( \mathcal{P}(A, B) \equiv \forall C \in D(\mathcal{C}(A, C) \rightarrow \mathcal{C}(B, C)) \).

RCC2-8 is described by the following eight JEPD relations: \( DC \) (DisConnected), \( EC \) (Externally Connected), \( PO \) (Proper overlap), \( EQ \) (Equal), \( TPP \) (Tangential proper Part), \( TPPc \) (TPP converse), \( NTPP \) (Non-tangential proper part), \( NTPPc \)
(NTPP converse). RCC-8 is a relation algebra which is associative and non-commutative. The RCC-8 relations are formally defined in point-set topology using axioms for Parthood (P) and Connectivity (C) as follows:

\[
\begin{align*}
DC(A, B) & \equiv \neg C(A, B) \\
EC(A, B) & \equiv C(A, B) \land (A^i \cap B^i = \emptyset) \\
PO(A, B) & \equiv (A^i \cap B^i = \emptyset) \land \neg P(A, B) \land \neg Pc(A, B) \\
EQ(A, B) & \equiv P(A, B) \land Pc(A, B) \\
NTPP(A, B) & \equiv PP(A, B) \land (\neg \exists C)[EC(C, A) \land EC(C, B)] \\
NTPPc(A, B) & \equiv PPc(A, B) \land (\neg \exists C)[EC(C, A) \land EC(C, B)] \\
TPP(A, B) & \equiv PP(A, B) \land (\exists C)[EC(C, A) \land EC(C, B)] \\
TPPc(A, B) & \equiv PPc(A, B) \land (\exists C)[EC(C, A) \land EC(C, B)]
\end{align*}
\]

The RCC-8 relations are depicted as the leaf nodes of a relation tree in Fig. 1. It is observed that EQ is reflexive, EQ, EC, and DC are symmetric, and EQ, NTPP, and NTPPc are transitive. In many situations, in addition to the relation between two objects in 2D (or 3D), the user also needs to know the relation between the projections of the objects in a 2D plane. We next discuss in detail the distinction between RCC-8 and VRCC-3D+ namely, how the VRCC-3D+ relations account for the various perspective projections onto a view plane in R³.

Relations \{DC, EC, PO, EQ, TPP, NTPP, TPPc, NTPPc\} have been proven to form a JEPD set, which is known as RCC-8. In some applications,⁹,²¹ it is more useful to simplify and combine some of these relations to get a smaller subset, RCC\(_k\) for \(k \in \{1, 2, 3, 5, 7\}\).
One of most well-known sets is RCC$_5$ or RCC-5. Here, the five relations are \{DR, PO, EQ, PP, PPc\} which contains DR (DiscRete), the combination of DC and EC; PO (Proper Overlap), EQ (Equal), PP (Proper Part), which is the combination of TPP and NTPP; and PPc (Proper Part converse), which is composed of TPPc and NTPPc. Similar sets of one, two, three, or five relations are known as RCC-1, RCC-2, RCC-3, RCC-5, RCC-7, and RCC-8, respectively:\(^2_{21}\):

- RCC-1 = \{S\} where S is disjunction of all eight relations,
- RCC-2 = \{O, DR\}, here DR is the disjunction of DC and EC; O is the disjunction of six relations \{PO, EQ, TPP, NTPP, TPPc, NTPPc\},
- RCC-3 = \{ONE, EQ, DR\}, here ONE is the disjunction of five relations \{PO, TPP, NTPP, TPPc, NTPPc\},
- RCC-5 = \{PP, PPc, PO, EQ, DR\}, here PP is the disjunction of TPP and NTPP; and PPc is the disjunction of TPPc and NTPPc,
- RCC-7 = \{DC, DR, P, Pc, NTPP, NTPPc, EQ\}, here P is disjunction of PP and PO; this is a slightly strange combination which is not JEPD,
- RCC-8 = \{DC, EC, PO, TPP, TPPc, NTPP, NTPPc, EQ\}.

RCC\(^2_{22}\)-23 extends RCC-8 to 23 relations in order to accommodate concave regions in 2D. This expanded set of relations is based on primitives for connection and convex hull (i.e. primitives for inside, partially inside, outside, and an inverse for each of those three relations). In addition to the PO, EQ, TPP, TPPc, NTPP, and NTPPc, relations of RCC-8, RCC-23 distinguishes eight relations for DC (based on the aforementioned convexity predicates), and nine relations for EC (also based on the convexity predicates).

RCC\(^2_{23}\)-62 is even more expressive than RCC-23, whereas RCC-23 considers a concave region as one whole part, RCC-62 decomposes such a region into an outside, boundary, interior, and inside. The resulting 62 relations are based on a 16-intersection that compares one object’s outside, boundary, interior, and inside with those of another object. However, like RCC-8 and RCC-23, RCC-62 only describes the relationship between regions considering two dimensions.

Although the QSR problem has been studied extensively in the literature\(^5\) the efforts mostly have been constrained to simply connected 2D regions. In contrast to RCC-8, RCC-23, and RCC-62, the LOS\(^{11-14}\) and ROC\(^6-20\) models qualify the spatial relation between 2D regions in terms of the obscuration that occurs between them. LOS-14 differs from ROC-20 in that it is restricted to objects that do not overlap in 3D. The 14 relations of LOS-14 describe the spatial relationship between two 2D objects in terms of whether or not one completely or partially hides the other using a qualitative depth relation between the objects; one object can be in front of another, but the two cannot be of equal depth. ROC-20 extends LOS-14 by allowing objects to be concave (and hence accommodates mutual occlusion, adding six relations to the LOS-14 set). ROC-20 defines all spatial relations in terms of a combination of occlusion (i.e. NonOccludes, PartiallyOccludes, MutuallyOccludes, and TotallyOccludes) and, as applicable, an RCC-8 relation.
In the aforementioned RCC-based models, the complete spatial relationship, not just the degree of obscuration, is only with respect to a particular 2D viewpoint. Over the past three years, the authors have investigated a different approach for spatial reasoning over 3D objects. An extensive discussion of those early models (named RCC-3D and VRCC-3D) is beyond the scope of this paper; the reader may refer to Ref. 24 for details. They subsequently were redesigned (and renamed VRCC-3D+) to resolve some ambiguities in the obscuration relations.

Each VRCC-3D+ relation takes the form $R_{\text{xObs}}y$ where $R$ represents the RCC-8 relation (computed in 3D), xObs refers to the type of obscuration in the projection plane (where $n\text{Obs} =$ no obscuration, $c\text{Obs} =$ complete obscuration, $p\text{Obs} =$ partial obscuration, and $e\text{Obs} =$ equal obscuration), and $y$, where applicable, refers to the qualitative distance the object from the viewer (where $c =$ converse, $e =$ equidistant). In our discussion, all the objects are assumed to be on one side of the plane relative to the view reference point. Not all combinations of RCC-8 relations and obscuration relations are physically possible. There are 37 VRCC-3D+ relations in all, as depicted by “*” in Table 1.

These relations can be refined by considering relative 3D distances as seen by the viewer. This is particularly critical in VRCC-3D+ for occlusion support where the distance of a pair of objects from the viewer is required to determine which object is closer to the viewer. In that case, the closer object is visible relative to the farther object, along with other intersection values.

2.3. Relative distance between 3D objects and its use in conjunction with perspective projections

Although the early RCC-3D and VRCC-3D models24 showed potential for studies involving 3D spatial data, it was determined that ambiguous analyses could occur in certain situations. As shown in Fig. 2, it was not possible to determine obscuration in 3D from only the 2D projections. That is, we could not say that, just because the projections intersected, the objects intersected in 3D space. More precisely, if we knew $R_1(A, B), R_2(B, C), \text{Obs}_1(A, B)$, and $\text{Obs}_2(B, C)$ (where $R_i$ is one of the spatial relations and Obs, is one of the obscuration relations), then we could only determine $R_3(A, C)$ and $\text{Obs}_3(A, C)$; we could not tell whether $A$ was in front of $C$, or $C$ was in front of $A$. In projections, the information about the distance from the projection plane was lost. In everyday life, we see the objects in 3D space. Occlusion computation is simplified by keeping track of qualitative relative distance, “A is closer to the viewer than B” is as opposed to quantitative distance, “how much closer (in meters) is A to B.” The relative distance determines the obscuring object, and the projection determines the possible type of obscuration. In Fig. 2, there are two locations of $B$ that give the same projection in the perspective projection plane. From the projections $A_P$ and $B_P$, the user cannot conclude anything without knowing the qualitative relative distance. If $A$ is in front of $B$, then $A$ partially obscures $B$. If $A$ is farther than $B$, then $A$ is completely obscured by $B$. This problem13 was resolved in
Table 1. The 37 VRCC3D+ relation indicated by "*". Each "*" is interpreted as $R_{x\text{Obs}_y}$.

<table>
<thead>
<tr>
<th>nObs</th>
<th>nObs_c</th>
<th>nObs_e</th>
<th>pObs1</th>
<th>pObs2</th>
<th>pObs_c1</th>
<th>pObs_c2</th>
<th>pObs_e</th>
<th>eObs</th>
<th>eObs_c</th>
<th>eObs_e</th>
<th>cObs</th>
<th>cObs_c</th>
<th>cObs_e1</th>
<th>cObs_e2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>EC</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>PO</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>EQ</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>TPP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>TPPc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NTPP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td>NTPPc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
VRCC-3D+ by considering a depth attribute, InFront, along with the three intersection attributes (IntInt, IntBnd, and BndInt) of projections of objects.

3. Efficiency of Intersection Model

3.1. 8-intersection versus 4-intersection

The eight axioms of RCC-8 (and hence VRCC-3D+) are intuitive from geometric visualization as shown in Fig. 1. They can be derived using weak connectedness and weak equality. Also, they can be defined in terms of the 9-intersection (or 8-intersection) framework. In fact, there are combinations of at most 4-intersections that are sufficient to replace the 9-intersections Model. Here we briefly describe how at most 4-intersections can replace 9-intersections for computational efficiency. This is a much more critical issue when the implementation is for 3D point data (as is the case for VRCC-3D+) rather than just 2D point data.

For a pair \((A, B)\) of 3D objects, the 9-intersection matrix defines nine intersections as shown in Table 2: \(A^i \cap B^i, A^i \cap B^b, A^i \cap B^e, A^b \cap B^i, A^b \cap B^b, A^b \cap B^e, A^e \cap B^i, A^e \cap B^b, A^e \cap B^e\), which can be written as predicates IntInt\((A, B)\), IntBnd\((A, B)\), IntExt\((A, B)\), BndInt\((A, B)\), BndBnd\((A, B)\), BndExt\((A, B)\), ExtInt\((A, B)\), ExtBnd\((A, B)\), ExtExt\((A, B)\), respectively.

Two regions may or may not intersect. If they intersect, it is qualitatively denoted by non-empty intersection \((\neg \emptyset)\) or predicate value T for true intersection. If they do not intersect, it is qualitatively denoted by empty intersection \((\emptyset)\) or predicate value F for no intersection. The complete table of true and false intersection values characterizing the RCC-8 relations is given in Table 3.

First, for bounded objects, we observe that intersection ExtExt\((A, B)\) is always non-empty. It is trivial because if \(A^e \cap B^e = \emptyset\), then \(A^e \subseteq \overline{B}\). Since \(A^e\) is unbounded and \(\overline{B}\) is bounded, it is impossible to have \(A^e \subseteq \overline{B}\). Hence, \(A^e \cap B^e = \neg \emptyset\). Thus, using 9-intersection is equivalent to 8-intersection without loss of any accuracy.
The next step is to further simplify the computations. We show that the 8-intersection is equivalent to the 4-intersections because we can derive the other four intersections from the knowledge of \(\text{IntInt}(A, B)\), \(\text{IntBnd}(A, B)\), \(\text{BndInt}(A, B)\), and \(\text{BndBnd}(A, B)\). In particular for DC and EC, from knowledge of \(\text{IntInt}(A, B)\), \(\text{BndBnd}(A, B)\), we can derive the remaining six intersections. Thus, computation of six intersections is not necessary in implementation, leading to 75\% gain in computation efficiency. Similarly, for NTPP and NTPPc, only \(\text{IntInt}(A, B)\), \(\text{BndBnd}(A, B)\), \(\text{IntBnd}(A, B)\), \(\text{BndBnd}(A, B)\) need to be computed, a savings of 62.5\% computation effort. For PO, EQ, TPP, TPPc, only \(\text{IntInt}(A, B)\), \(\text{BndBnd}(A, B)\), \(\text{IntBnd}(A, B)\), \(\text{BndBnd}(A, B)\) can be used, saving 50\% of the computation effort. On the average, using at most 4-intersections saves 60\% of the computation effort. All of the 2-intersection, 3-intersection, and 4-intersection predicate analyses are synthesized in Fig. 3, confirming that rather than 8-intersections, we need only 2-intersections for DC and EC, 3-intersections for NTPP and NTPPc, and 4-intersections for PO, EQ, TPP, and TPPc. In Fig. 3, if two objects intersect, the predicate value is T for true, otherwise, the predicate is F for false.

Herein, we only give formal analysis for one case, the 4-intersection for PO (i.e. the last row in Table 3). We show that 4-intersections are equivalent to a first order logic axiomatic framework for RCC-8 relations and the 9-intersection model. All the other cases are fully discussed in Sabharwal-Leopold.4

Recall, if \(d\) is a distance metric, for \(x \in R^n\), the neighborhood of \(x\) with radius \(r > 0\) is defined as a set \(\{y : d(x, y) < r \text{ for } r > 0\}\); it is denoted by \(N(x, r)\). When the radius is understood and its value is immaterial in proofs, we simply use the qualitative neighborhood, \(N(x)\). An element \(x\) is an interior point of \(A\) if there is a neighborhood \(N(x)\) of \(x\), such that \(N(x) \subseteq A\). An element \(x\) is an exterior point of \(A\) if there is a neighborhood \(N(x)\) of \(x\), such that \(N(x) \subseteq A^e\). If \(x \in A^i\), then every \(N(x)\) intersects the interior and exterior of \(A\) (i.e. \(N(x) \cap A^i = \emptyset\) and \(N(x) \cap A^e = \emptyset\)).

---

### Table 2. 9-intersection matrix.

<table>
<thead>
<tr>
<th></th>
<th>(A^i \cap B^i)</th>
<th>(A^i \cap B^b)</th>
<th>(A^i \cap B^c)</th>
<th>(A^b \cap B^i)</th>
<th>(A^b \cap B^b)</th>
<th>(A^b \cap B^c)</th>
<th>(A^c \cap B^i)</th>
<th>(A^c \cap B^b)</th>
<th>(A^c \cap B^c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>EC</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>NTPP</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>NTPPc</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>EQ</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>TPP</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>TPPc</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>PO</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>

The next step is to further simplify the computations. We show that the 8-intersection is equivalent to the 4-intersections because we can derive the other four intersections from the knowledge of \(\text{IntInt}(A, B)\), \(\text{IntBnd}(A, B)\), \(\text{BndInt}(A, B)\), and \(\text{BndBnd}(A, B)\). In particular for DC and EC, from knowledge of \(\text{IntInt}(A, B)\), \(\text{BndBnd}(A, B)\), we can derive the remaining six intersections. Thus, computation of six intersections is not necessary in implementation, leading to 75\% gain in computation efficiency. Similarly, for NTPP and NTPPc, only \(\text{IntInt}(A, B)\), \(\text{BndBnd}(A, B)\), \(\text{IntBnd}(A, B)\), \(\text{BndBnd}(A, B)\) need to be computed, a savings of 62.5\% computation effort. For PO, EQ, TPP, TPPc, only \(\text{IntInt}(A, B)\), \(\text{BndBnd}(A, B)\), \(\text{IntBnd}(A, B)\), \(\text{BndBnd}(A, B)\) can be used, saving 50\% of the computation effort. On the average, using at most 4-intersections saves 60\% of the computation effort. All of the 2-intersection, 3-intersection, and 4-intersection predicate analyses are synthesized in Fig. 3, confirming that rather than 8-intersections, we need only 2-intersections for DC and EC, 3-intersections for NTPP and NTPPc, and 4-intersections for PO, EQ, TPP, and TPPc. In Fig. 3, if two objects intersect, the predicate value is T for true, otherwise, the predicate is F for false.

Herein, we only give formal analysis for one case, the 4-intersection for PO (i.e. the last row in Table 3). We show that 4-intersections are equivalent to a first order logic axiomatic framework for RCC-8 relations and the 9-intersection model. All the other cases are fully discussed in Sabharwal-Leopold.4

Recall, if \(d\) is a distance metric, for \(x \in R^n\), the neighborhood of \(x\) with radius \(r > 0\) is defined as a set \(\{y : d(x, y) < r \text{ for } r > 0\}\); it is denoted by \(N(x, r)\). When the radius is understood and its value is immaterial in proofs, we simply use the qualitative neighborhood, \(N(x)\). An element \(x\) is an interior point of \(A\) if there is a neighborhood \(N(x)\) of \(x\), such that \(N(x) \subseteq A\). An element \(x\) is an exterior point of \(A\) if there is a neighborhood \(N(x)\) of \(x\), such that \(N(x) \subseteq A^e\). If \(x \in A^i\), then every \(N(x)\) intersects the interior and exterior of \(A\) (i.e. \(N(x) \cap A^i = \emptyset\) and \(N(x) \cap A^e = \emptyset\)).
Only four predicates, IntInt, BndBnd, IntBnd, and BndInt, (which are shaded in Table 4) are needed for PO, EQ, TPP, and TPPc. The other four of the 8-intersection predicates can be derived mathematically. Here, we provide proofs only for the case of PO (depicted in Fig. 4), which corresponds to the fourth row in Table 4.

Given: \( A^i \cap B^i = \emptyset, A^b \cap B^b = \emptyset \),

\[
A^i \cap B^b = \emptyset, \quad A^b \cap B^i = \emptyset \tag{3.1}
\]

To show \( A^i \cap B^e = \emptyset \),

\[
\tag{3.2}
\]

Now from (3.1), \( A^i \cap B^b = \emptyset \).
Let \( x \in A^i \cap B^b \). Then there is a neighborhood \( N(x) \) of \( x \) such that \( N(x) \subseteq A^i \).
Since \( x \in B^b \), every neighborhood \( N(x) \) of \( x \) intersects \( B^i \) and \( B^e \); so that \( N(x) \cap B^e = \emptyset \).
Hence, \( A^i \cap B^e = \emptyset \).

To show \( A^e \cap B^i = \emptyset \),

\[
\tag{3.3}
\]

Table 4. 4-intersection predicates (shaded) needed to compute EQ, TPP, TPPc, and PO.

<table>
<thead>
<tr>
<th></th>
<th>IntInt</th>
<th>BndBnd</th>
<th>IntBnd</th>
<th>BndInt</th>
<th>IntExt</th>
<th>BndExt</th>
<th>ExtInt</th>
<th>ExtBnd</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>TPP</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>TPPc</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>PO</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
</tbody>
</table>
By interchanging $A$ and $B$ in the above argument for (3.2) and symmetry, we can infer that $A^e \cap B^i = \emptyset$.

To show $A^b \cap B^e = \emptyset$,

$$A^b \cap B^e = A^b \cap (B^i \cup B^b)^c = A^b \cap (B^{ic} \cap B^{bc})$$

Now $A^b \cap B^e = A^b \cap B^i = \emptyset$, because 

if $A^b \cap B^{ic} = \emptyset$, 

$A^b \subseteq B^i$, which means $A^b \cap B^b = \emptyset$.

That is a contradiction to (3.1) given that $A^b \cap B^i = \emptyset$.

We also know that $A^b \cap B^{bc} = \emptyset$; trivially $A^b \cap B^{bc} = A^b \cap (B^i \cup B^e) = \emptyset$ because (3.1) asserts that $A^b \cap B^i = \emptyset$.

Hence, $A^b \cap B^e$ is non-empty.

To show $A^e \cap B^b = \emptyset$,

$$A^e \cap B^b = \emptyset$$

From symmetry, and by interchanging $A$ and $B$ in the above argument (3.4), we have $A^e \cap B^b = \emptyset$.

Here we provide proofs using one case (PO) for 4-intersection, to prove the axiomatic definitions of RCC-8 relations. Hence, they are no longer axioms, but may be treated as theorems.

Given the 4-intersections

$$A^i \cap B^i = \emptyset, A^b \cap B^b = \emptyset,$$

$$A^i \cap B^b = \emptyset$$

we show that

PO($A, B$) $\equiv$ $(A^i \cap B^i = \emptyset) \land \neg P(A, B) \land \neg Pc(A, B)$.

Specifically, we prove that

$(A^i \cap B^i = \emptyset) \land \neg P(A, B) \land \neg Pc(A, B)$ holds.

(1) We have proved in (3.2) and (3.3) that

$A^i \cap B^e = \emptyset$, and $A^e \cap B^i = \emptyset$.

Since $A^i \cap B^e = \emptyset$, it is not true that $A^i \subseteq B$; therefore, $\neg P(A, B)$.
Since \( A^e \cap B^i = \neg \emptyset \), it is not true that \( B^i \subseteq A \);
therefore, \( \neg P(B, A) \) (i.e. \( \neg P_c(A, B) \)).

(2) We are given in (3.1) that \( A^i \cap B^i = \emptyset \);
therefore from (1) and (2) we have
\[
(A^i \cap B^i) = \neg \emptyset \land \neg P(A, B) \land \neg P_c(A, B).
\]

Hence, we have shown that the 4-intersections satisfy the criteria for \( PO(A, B) \).
This shows that for computational efficiency, we need to compute only four
intersections instead of nine intersections. Also, using symmetry, the two algorithms
for \( \text{IntBnd} \) and \( \text{BndInt} \) can be replaced with single algorithms. So we will need to
implement only three intersection algorithms instead of nine.

4. Composition of Spatial Relations

We next discuss composition-based reasoning. This is a critical component for per-
forming QSR, and becomes a computationally intractable problem as the number of
relations in the region connection calculi increases (e.g. VRCC-3D+ has 37 relations
as compared to RCC-8 which has 8 relations).

For the set \( D \), let \( R_1 \) and \( R_2 \) be relations on \( D \). The composition of relations \( R_1 \)
and \( R_2 \) is denoted by \( R_1 \circ R_2 \), \( R_1 \circ R_2 \subseteq D^2 \), and is defined by
\( R_1 \circ R_2 = \{(A, C) \in D^2 : \exists B \in D \exists (A, B) \in D^2, \text{ and } (B, C) \in D^2 \} \).
In spatial reasoning, the instantiation of \( B \) is sufficient, but not necessary, in the definition of composition.

The notion of \textit{weak composition} is defined as follows. Let \( R = \{R_1, R_2, \ldots, R_n\} \) be
a finite set of JEPD binary relations on a domain \( D \). The weak composition is denoted
by \( \circ_w \) and defined as \( \circ_w : \mathcal{R} \times \mathcal{R} \rightarrow 2^\mathcal{R} \), meaning for given \( A, B, C \in D \),
with \( R_i(A, B) \land R_j(B, C) \), \( \exists k \in \{1, 2, \ldots, n\} \) such that \( R_i \circ_w R_j(A, C) = R_k(A, C) \) so that
\( R_i(A, B) \land R_j(B, C) \land R_k(A, C) \) is true for some \( k \). The problem then is to determine
whether there actually is an instantiation \( A, B, C \ni R_i(A, B) \land R_j(B, C) \land R_k(A, C) \).

Note that, weak composition is not a single base relation, but rather a disjunction
of a set of relations. That is, for specific instantiation of relations \( R_i \) and \( R_j \) in \( B \) with
objects \( A, B, \) and \( C \), for which \( R_i(A, B) \) and \( R_j(B, C) \) hold, it is the case that
\( R_i \circ_w R_j(A, C) \in \{R_1(A, C), R_2(A, C), \ldots, R_n(A, C)\} \). More precisely, \( R_i \circ_w R_j \) is the
disjunction of the \textit{smallest subset} of \( \{R_1, R_2, \ldots, R_n\} \) for which the disjunctive
identity \( R_i \circ_w R_j = R_1 \lor R_2 \lor \ldots \lor R_n \) holds. For spatial relations, the weak
composition can be expressed as the disjunction of base relations in the set
\( \{R : R \in B, R \cap (R_i \circ R_j) \neq \emptyset \} \). Thus, the \textit{mathematical} composition \( \circ \) and spatial
relation \textit{weak composition} \( \circ_w \) are related by the equation:
\[
R_i \circ_w R_j = \{R : R \in \mathcal{R}, R \cap (R_i \circ R_j) \neq \emptyset \}.
\]

The composition of two spatial relations can be a single relation in some cases, e.g.
in RCC-8 the composition \( \text{TPP}_0 \circ \text{NTPP} \) is \( \text{NTPP} \) (see Table 5(a)). In some cases, it
can be the disjunction of several relations depending on the objects under
consideration. Dependent on object instantiation, \( ECo_w TPc \) is one of \{DC, EC\}, e.g. DR (see Table 5(b). Based on relation algebra terminology, composition is a disjunction of relations independent of the object under consideration, in fact, \( ECo_w EC = DC \lor EC \lor PO \lor EQ \lor TPP \lor TPPc \) (see the graphical representation in Fig. 5). Note that, \( DCo_w DC \) is the set of all base relations; known as the universal relation, which is denoted by an asterisk, \(*\), or shading of all nodes in the node tree (as shown in Table 6). It amounts to saying, it is the most ambiguous case, no information is inferred, it can be anything. Looking at the composition table (Table 6), we see that there are 27 singletons, 3 universal, and the rest of the 34 entries are 2 to 6 base relations. Singleton composition is also called Crisp composition.\(^{24}\) Examples of low resolution logics composition \( 2 \times 2 \) composition tables are given in Tables 5(a) and 5(b).

4.1. Composition table: Construction for base relations

There are several methods for creating composition tables. One way is to manually create a static table.\(^{25}\) Although several algorithms\(^{26,27}\) have been proposed for automated construction, they can produce erroneous results, and can be quite

<table>
<thead>
<tr>
<th></th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPP</td>
<td>PP</td>
</tr>
<tr>
<td>NTPP</td>
<td>NTPP</td>
</tr>
</tbody>
</table>

Table 5. RCC-8 partial composition tables.

(a) PPO\(_w\)PP

<table>
<thead>
<tr>
<th></th>
<th>PP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPP</td>
<td>PP</td>
</tr>
<tr>
<td>NTPP</td>
<td>NTPP</td>
</tr>
</tbody>
</table>

(b) DRO\(_w\)PPc

<table>
<thead>
<tr>
<th></th>
<th>PPc</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPPc</td>
<td>NTPPc</td>
</tr>
</tbody>
</table>

Fig. 5. Representation composition \( ECo_w EC \) for all possible values of \( C \) in \( EC(A, B)_{o_w} EC(B, C) \).
Table 6. Composition table of RCC-8 relations. Composition table entry is an asterisk "*" for disjunction of all relations in \( B \).

<table>
<thead>
<tr>
<th>( R(A,B) )</th>
<th>o</th>
<th>DC</th>
<th>EC</th>
<th>PO</th>
<th>TPP</th>
<th>NTPP</th>
<th>TPPc</th>
<th>NTPPc</th>
<th>EQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>*</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
</tr>
<tr>
<td>EC</td>
<td>DR,PO,PP</td>
<td>DR,PO,PP</td>
<td>DR,PO,PP</td>
<td>DR,PO,PP</td>
<td>DR,PO,PP</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>EC</td>
</tr>
<tr>
<td>PO</td>
<td>DR,PO,PC</td>
<td>DR,PO,TPP,TPPC,EQ</td>
<td>DR,PO,PP</td>
<td>EC,PO,PP</td>
<td>PO,PP</td>
<td>PO,PP</td>
<td>DR,PO,PP</td>
<td>PO</td>
<td></td>
</tr>
<tr>
<td>TPP</td>
<td>DC</td>
<td>DR</td>
<td>DC</td>
<td>DR,PO,PP</td>
<td>PP</td>
<td>NTPP</td>
<td>DR,PO,PPc,EQ</td>
<td>DR,PO,PPc</td>
<td>TPP</td>
</tr>
<tr>
<td>NTPP</td>
<td>DC</td>
<td>DC</td>
<td>DC</td>
<td>DR,PO,PP</td>
<td>NTPP</td>
<td>NTPP</td>
<td>DR,PO,PPc</td>
<td>*</td>
<td>NTPP</td>
</tr>
<tr>
<td>TPPc</td>
<td>DR,PO,PPc</td>
<td>EC,PO,PPc</td>
<td>PO,PPc</td>
<td>PO,TPP,TPPC,EQ</td>
<td>PO,PP</td>
<td>PPc</td>
<td>NTPPc</td>
<td>TPPc</td>
<td></td>
</tr>
<tr>
<td>NTPPc</td>
<td>DR,PO,PPc</td>
<td>PO,PPc</td>
<td>PO,PPc</td>
<td>PO,PPc</td>
<td>PO,PP,PPPC,EQ</td>
<td>NTPPc</td>
<td>NTPPc</td>
<td>NTPPc</td>
<td></td>
</tr>
<tr>
<td>EQ</td>
<td>DC</td>
<td>EC</td>
<td>PO</td>
<td>TPP</td>
<td>NTPP</td>
<td>TPPc</td>
<td>NTPPc</td>
<td>EQ</td>
<td></td>
</tr>
</tbody>
</table>
complicated to implement. A recently presented method based on machine learning
techniques\textsuperscript{8} is semi-automatic, but imposes some very severe restrictions on the
construction of the table. Unfortunately, the results are not guaranteed to be 100% accurate, and the method assumes that the input dataset is a prior consistent. The
primary advantage of the method is that the dynamic table is local to the problem context, and consequently more efficient to use. A static table can be too complex if it
must cover all cases for completeness, and its use may impede the overall efficiency of the
reasoner application.

It is often the case that a significant amount of knowledge can be deduced from a
small set of binary relations. The inferences can be obtained by using composition-
based reasoning. The relation semantics is used to construct the composition table to
verify the correctness and completeness of inferences. In general, for composition
$R_1 \circ R_2$, we would check $R_1(a, b) \land R_2(b, c) \land R_k(a, c)$ for $n$ values of $k$ for consistency.
If there are $n$ relations, the composition table will have $n^2$ entries. Each table entry is
represented by more general relations, specifically the disjunction of at most $n$ base
relations, requiring $n$ consistency checks. Thus, the table construction is of order
$O(n^3)$. In practice, for composition $EQ_0 \circ EQ$ there is only one consistency check
instead of eight. For composition $TPP_0 \circ TPP$, only two consistency checks are
necessary instead of eight. For composition $EC_0 \circ EC$, there are six consistency checks
instead of eight (see Fig. 5). $DC_0 \circ DC$, $PO_0 \circ PO$ and $NTPP_0 \circ TPP_c$ are the dis-
junction of all relations. There is a large amount of duplicated effort even in the
construction of the composition table $O(n^3)$.

Historically, composition has been used for automated reasoning in a variety of
applications. It has been used for temporal reasoning,\textsuperscript{28} as well as spatial reasoning.\textsuperscript{2,3}
But when the set of relations is large, it is difficult to handcraft a large composition
table. An architecture for reasoning based on composition tables will need to con-
struct tables dynamically due to the nature of the problem and the number of relations
required.

In the RCC-8 composition table shown in Table 6 (using 9-intersection,\textsuperscript{25}) the left
column relations are for object pair $(A, B)$, the top row corresponds to object pair
$(B, C)$, and the table entries correspond to object pair $(A, C)$.

A Prolog program was written to generate a composition table (CT) for VRCC-
3D (and, subsequently, VRCC-3D\textsuperscript{+}), providing the ability to answer questions such as, “given three objects, $A, B,$ and $C$, and knowledge of relations $R_1(A, B)$, and
$R_2(B, C)$, what can be said about the relation between $A$ and $C$.” A prototype
VRCC-3D reasoner was then implemented, and tested to compare automated de-
termination of spatial relationships between 3D reconstructions of anatomy (in OBJ
file format) against the determination of experts who had examined the original
specimens manually. The preliminary results were very promising, and work com-
menced to create an application that would allow the user to check for inconsistency
in relations between abstract temporal “states” of the objects, and determine possible
relations that would have had to occur to transition from one state to another. The
integration of that functionality together with a visual user interface was named VRCC-3D.\textsuperscript{24}

### 4.2. Composition table: Applications

An important feature of the RCC models is that the base relations are JEPD. The set of possible spatial relations between two objects is determined by using relation composition. Thus, the composition table is useful for automated reasoning where identical calculations are performed repeatedly. Instead of reinventing the computations, they are pre-computed once and stored in the composition table. This table is used as a lookup table to avoid expensive 3D object intersection computations. The composition table is a central part of any composition-based reasoning.

Interesting algebras are based on finite sets of JEPD relations, which are called basic, base, and atomic relations. They represent unambiguous knowledge with respect to a level of granularity. But disjunctions represent ambiguous knowledge. If the relations are JEPD, closed under relevant operations, a constraint based method can be applied. The composition table of basic relations and a way to compose complex relations is needed to efficiently perform reasoning. Without the JEPD property, reasoning over the theory potentially would be undecidable, and the relations could be ambiguous. The JEPD relations, the composition table, and a path-completion algorithm\textsuperscript{29} not only enable reasoning over these theories, but also can be used to detect inconsistency. The set of JEPD relations defined for VRCC-3D\textsuperscript{+} was influenced by the desire to reason over 3D objects.

### 4.3. Composition table: Complexity

Each composition relation is associated with three objects, say $A$, $B$, and $C$, which can be represented as the vertices of a triangular graph. The relations then can be associated with the edges of the triangle (i.e. $R_1(A,B)$, $R_2(B,C)$, and $R_3(A,C)$). The complexity of computing the composition table is proportional to the number of triangles for all such compositions. If a relation is symmetric, the associated edge will be undirected. If a relation is asymmetric, the corresponding edge will be directed. If a triangle is rotated or inverted, an equivalent configuration results.

Let $(A, B, C)$ be the vertices of a triangular graph with edge direction represented by arrows along the edges $AB, BC, CA$. We use $u$ for (undirected) edges corresponding to symmetric relations, and $v, w$ for (directed) edges corresponding to asymmetric relations. The possible triangles may have no directed edges as in $(u, u, u)$, one directed edge as in $(w, u, u)$, two directed edges as in $(u, v, w)$, $(u, w, v)$, and $(u, v, v)$, or three directed edges as in $(v, v, v)$ and $(w, v, v)$; these different cases are depicted in Fig. 6. If $s = \#u$ (symmetric relations), and $a = \#v + \#w$ (asymmetric relations), then $n = s + 2a$. Let $T$ be the total number of triangles. After calculating the possible instantiations, from Bennet,\textsuperscript{30} we have

$$T = (1/6)(s^3 + 3s^2 + 2s) + s^2a + s(2a^2 + a) + (1/3)(4a^3 + 2a).$$
Since \( s = n - 2a \), we know \( T = (1/6)(n^3 + 3n^2 + 2n) - n^a \). Therefore, the number of distinct triangles reflects about one sixth of the brute force approach total, \( n^3 \).

Utilizing these triangle criteria in the composition table construction reduces the computation effort considerably, as shown in Table 7 for various region connection calculi. Here the percentage of computational effort (i.e. the rightmost column in Table 7) is calculated as \(((1/6)(n^3 + 3n^2 + 2n) - na)/n^3*100\). This table shows that the computation effort is at most 24%, a savings of at least 76% over brute force methods.

<table>
<thead>
<tr>
<th>System</th>
<th>( s )</th>
<th>( a )</th>
<th>( n^3 )</th>
<th>( T )</th>
<th>( % )</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCC-5</td>
<td>3</td>
<td>1</td>
<td>125</td>
<td>30</td>
<td>24</td>
</tr>
<tr>
<td>RCC-7</td>
<td>3</td>
<td>2</td>
<td>343</td>
<td>70</td>
<td>20</td>
</tr>
<tr>
<td>RCC-8</td>
<td>4</td>
<td>2</td>
<td>512</td>
<td>104</td>
<td>20</td>
</tr>
<tr>
<td>IC-13</td>
<td>1</td>
<td>6</td>
<td>2197</td>
<td>377</td>
<td>17</td>
</tr>
<tr>
<td>LOS-14</td>
<td>2</td>
<td>6</td>
<td>2744</td>
<td>476</td>
<td>17</td>
</tr>
<tr>
<td>VRCC-3D</td>
<td>5</td>
<td>6</td>
<td>4913</td>
<td>867</td>
<td>18</td>
</tr>
<tr>
<td>ROC-20</td>
<td>6</td>
<td>7</td>
<td>8000</td>
<td>1400</td>
<td>18</td>
</tr>
<tr>
<td>ROC-23</td>
<td>7</td>
<td>8</td>
<td>12167</td>
<td>2116</td>
<td>17</td>
</tr>
<tr>
<td>VRCC-3D+</td>
<td>9</td>
<td>14</td>
<td>50653</td>
<td>8621</td>
<td>17</td>
</tr>
<tr>
<td>RCC-62</td>
<td>10</td>
<td>26</td>
<td>238328</td>
<td>40052</td>
<td>17</td>
</tr>
</tbody>
</table>
4.4. Divide-and-conquer construction algorithm for VRCC-3D+
composition table

In VRCC-3D+, there are 37 relations. Clearly, it is not practical to create the VRCC-
3D+ composition table manually (i.e. each entry is a disjunction of at most 37
relations). This also would be problematic for other mathematical models used for
QSR (e.g. RCC-62, which contains 62 relations). Here, we present an automated
construction approach that utilizes a divide-and-conquer strategy. We first factor out
the common attribute in the relations, then create smaller composition tables,
eventually integrating them into one overall table.

We start with a simple example to illustrate this. The RCC-8 composition table is
an 8 \times 8 table, where each entry is the disjunction of at most eight relations. Al-
ternately, it can be thought of as a (smaller) 4 \times 4 table, where each entry is a 2 \times 2
table whose entries are the disjunction of at most four relations. This requires
decomposing the relations into sub-categories (or equivalence classes) before creating
a full scale composition table. We also may take advantage of the intrinsic properties
of the relations to reduce the size of the table. For example, it may be observed that
composition of EQ with any relation is the relation itself (e.g. $X_0 \wedge EQ = EQ \wedge X$ for any relation X). This reduces the computation of an 8 \times 8 table to a 7 \times 7 table.
Also, once $PP_0 \wedge PP$ is computed, we do not need to compute the table for $PP_0 \wedge PPc$. Those entries are simply the converses of the entries in the table for $PP_0 \wedge PP$. Since
composition is not commutative, $PP_0 \wedge PPc$ is not the same as $PP_0 \wedge PPc$.

If we let $DR = \{DC, EC\}$, $PP = \{TPP, NTPP\}$, $PPc = \{TPPc, NTPPc\}$, and
$O = \{PO, EQ\}$, we get a set of four equivalence classes, $\{DR, O, PP, PPc\}$. If we take
advantage of the intrinsic property of EQ, O can be replaced with singleton
$O = \{PO\}$. This will yield a 4 \times 4 table whose entries correspond to sub-tables of size
$2 \times 2, 2 \times 1, 1 \times 2$, and $1 \times 1$ for $PP_0 \wedge PP$, $PP_0 \wedge PO$, $PO_0 \wedge PP$ and $PO_0 \wedge PO$, re-
spectively. This smaller composition table is shown in Table 8, where entries of *
denote all possible relations.

We provide two examples to illustrate how this “reduced” composition table can
be used to construct the full table. First, consider the composition $PP_0 \wedge PP = PP$,
where PP represents $\{TPP, NTPP\}$. $PP_0 \wedge PP$ can be expanded into a 2 \times 2 composition sub-table by creating the composition of four pairs whose entries come from
$\{TPP, NTPP\}$. Table 9 shows that the $PP_0 \wedge PP$ is a representative of the 2 \times 2 table
with entries from $\{TPP, NTPP\}$.

<table>
<thead>
<tr>
<th>o</th>
<th>DR</th>
<th>PO</th>
<th>PP</th>
<th>PPc</th>
<th>EQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>*</td>
<td>DR,PO,PP</td>
<td>DR,PO,PP</td>
<td>DR</td>
<td>DR</td>
</tr>
<tr>
<td>PO</td>
<td>DR,PO,PP</td>
<td>*</td>
<td>PO,PP</td>
<td>DR,PO,PPc</td>
<td>PO</td>
</tr>
<tr>
<td>PP</td>
<td>DR</td>
<td>DR,PO,PP</td>
<td>PP</td>
<td>*</td>
<td>PP</td>
</tr>
<tr>
<td>PPc</td>
<td>DR,PO,PPc</td>
<td>PO,PPc</td>
<td>PO,PP,PPc,EQ</td>
<td>PPc</td>
<td>PPc</td>
</tr>
<tr>
<td>EQ</td>
<td>DR</td>
<td>PO</td>
<td>PP</td>
<td>PPc</td>
<td>EQ</td>
</tr>
</tbody>
</table>
As a second example, consider $\text{DRo}_{\text{PPc}} = \text{DR}$, where $\text{DR}$ represents the set \{DC, EC\} and $\text{PPc}$ is the set \{TPPc, NTPPc\}. $\text{DRo}_{\text{PPc}}$ can be expanded into a 2 x 2 composition sub-table by creating the compositions of four pairs whose entries come from \{DC, EC\}. The composition of four pairs easily can be computed and integrated with $\text{DRo}_{\text{PPc}}$. Table 10 shows that $\text{DCo}_{\text{PPc}}$ is in fact a 2 x 2 table with entries from \{DC, EC\}.

Similarly, other composition sub-tables for RCC-8 can be computed and integrated to form the full 8 x 8 table. The utility of this strategy lies in handling cases where there are many relations, resulting in a large table. The ingenuity lies in partitioning the large set of relations into smaller subsets, a task that can be automated using various machine learning algorithms.

Now we apply this strategy to VRCC-3D+, which defines 37 relations. Each VRCC-3D+ relation consists of one of eight connectivity relations and one of 12 obscuration relations. Using the same strategy discussed above, we separately create the composition tables for the eight relations (previously shown in Table 6 and for the 12 obscuration relations (not shown). The two tables are then integrated to form the composite table covering all 37 relations. The integration of those composition tables can be performed using the following algorithm:

1. Compute composition table CT _RCC8;
2. Compute composition table CT _Obs12;
3. For each pair R1 _Obs1 and R2 _Obs2 in VRCC-3D+ do:
   4. For each R in CT _RCC8[R1, R2] do:
      5. Let ObsR be the set of obscuration relations applicable to R (see Table 3);
      6. For each xObs _y in

---

Table 9. Composition sub-table for $\text{PPo}_{\text{PP}}$.

<table>
<thead>
<tr>
<th>PP</th>
<th>TPP</th>
<th>NTPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>TPP</td>
<td>PP</td>
</tr>
<tr>
<td>NTPP</td>
<td>NTPP</td>
<td>NTPP</td>
</tr>
</tbody>
</table>

Table 10. Composition sub-table for $\text{DRo}_{\text{PPc}}$.

<table>
<thead>
<tr>
<th>PPc</th>
<th>TPPc</th>
<th>NTPPc</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>DC</td>
<td>DC</td>
</tr>
<tr>
<td>EC</td>
<td>DR</td>
<td>DC</td>
</tr>
</tbody>
</table>
Table 11. One entry from composition table for VRCC-3D+ connectivity relations.

<table>
<thead>
<tr>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
</tr>
<tr>
<td>{DC,EC,PO,TPPc,NTPPc}</td>
</tr>
</tbody>
</table>

Table 12. One entry from composition table for VRCC-3D+ occlusion relations.

<table>
<thead>
<tr>
<th>nObs</th>
</tr>
</thead>
<tbody>
<tr>
<td>pObs</td>
</tr>
<tr>
<td>{nObs, pObs, cObs}</td>
</tr>
</tbody>
</table>

Table 13. One entry from complete VRCC-3D+ composition table.

<table>
<thead>
<tr>
<th>DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>nObs</td>
</tr>
<tr>
<td>EC</td>
</tr>
<tr>
<td>pObs</td>
</tr>
<tr>
<td>{DC _nObs, DC _pObs, DC _cObs, EC _nObs, EC _pObs, EC _cObs, NTPPc _cObs, PO _cObs, PO _pObs, TPPc _cObs}</td>
</tr>
</tbody>
</table>

CT Obs12[Obs1, Obs2] \cap ObsR do:
Add R \_xObs \_y to
CT[R1 \_Obs1, R2 \_Obs2] ;
EndFor
EndFor
EndFor.

Due to space limitations, it is not possible to display the 37 \times 37 VRCC-3D+ composition table in its entirety. As an example of the construction, Tables 11–13 show one entry from each of (i) the 8 \times 8 composition table of VRCC-3D+ connectivity relations, (ii) the 12 \times 12 composition table of VRCC-3D+ obscuration relations, and (iii) the 37 \times 37 full composition table for VRCC-3D+ composite connectivity and occlusion relations, respectively.

5. Conceptual Neighborhood Graph

5.1. Conceptual neighbors

"Everything is related to everything else, but nearby things are more related than distant things"

Tobler’s First Law of Geography, 1979
Qualitative distances cannot be expressed by conventional measures. Most differentiating measures are derived from observation and experience in an *ad hoc* manner. Distance between two relations is computed by using some metric defined for the presentation of the relation qualitatively or quantitatively. The results are cognitively plausible only if they match the user’s concept of distance.

Computational distance of spatial objects and relations is based on geometric and non-geometric attributes. Geometric attributes are associated with object dimensions, whereas non-geometric attributes are non-numeric, categorical attributes such as near, far, color, shape, etc. Sometimes all attributes are treated homogeneously, whereas at other times the individual attributes may be weighted differently. The weights may be assigned to indicate the salience of the attributes consistent with visualization and human perception.

Conceptually two objects are neighbors if the topological distance between them is minimal. Two relations between pairs of objects are conceptual neighbors if the objects can be topologically continuously deformed from one relation to the other relation without encountering any other relation state. The 9-intersection characterization is useful for creating a composition table, but not suitable for calculating the conceptual distance. For example in Fig. 7, the topological distances for the RCC-8 relations computed based upon the 9-intersection model are not a good measure of perceived distance. The distance between PO and EQ is 6, and they are conceptual neighbors (as they can be deformed into each other without encountering any other relation in between). However, despite the fact that the distance between NTPP and NTPPc is smaller, 2, those two relations are not neighbors using the same criteria. In Fig. 8, the distances are calculated using the 4-intersection. Now the same edges (PO, EQ), (NTPP, NTPPc) are both of length 2, but PO and EQ are conceptual neighbors, and NTPP and NTPPc still are not conceptual neighbors. In Figs. 7 and 8, solid lines denote “smooth transitions”, whereas the dotted lines represent non-smooth

![Diagram](image)

**Fig. 7.** Partial graph of RCC-8 topological relation distances based on the 9-intersection model.
transition edges. It is important to distinguish the edges based on this “conceptual” distance, not just the absolute “metric” distance.

5.2. Conceptual neighborhood graph

A conceptual neighborhood graph should give a visual interpretation of qualitative distances between pairs of relations. A spatial relation graph is a graph whose nodes are relations, and an edge represents a neighbor relation. A set of relations is a conceptual neighborhood (graph) if there is a sequence of conceptual neighbors between every pair of relations. The relation changes gradually when there is a continuous topological deformation in the objects. Even in Fig. 12, the relation EC is closer to PO than DC. The conceptual neighborhood graph should retain only those edges that connect “conceptually” closer relations and prune the other edges.

The term transition is also vaguely used for conceptual neighbors. More precisely, let a pair of objects be continuously deformed from relation $R_1$ state to another relation $R_2$ state. This change in state is a direct transition if there is no other relation between $R_1$ and $R_2$. We call it a transition path if there is a sequence of direct transition relations between $R_1$ and $R_2$. This leads to the concept of “smooth” transitions.

6. Smooth Transitions

Topological relations permit topological transformations on regions, such as translation, rotation, and scaling in 3D. Each topological relation has a converse so that the basic relations are JEPD.

In RCC-8, there are eight atomic (basic or base) relations in the set $B$ such that between any two regions, there is one and only one relation between them from the

![Partial graph of RCC-8 topological relation distances based on the 4-intersection model.](image)
set $B$ of base relations. The set $E$, a set of smooth transition edges, represents deformations on relations due to the continuous topological transformation$^{31}$ of one or both objects. A conceptual neighborhood graph (CNG) is a graph $(B, E)$ that describes spatial configuration, where nodes are relations, and edges are smooth transitions. Since each relation has a converse, the graph is bidirectional. Each node is also a self loop, indicating the possibility that no significant change occurred. For simplicity, these are not marked in the graph.

In Figs. 9–11, the configuration from DC to EC, EC to PO is obtained by translating/scaling the objects. Specifically, the relations change from DC to EC to PO as follows:

(a) Translating only one object: translation of $A$ or $B$ (Fig. 9).
(b) Scaling only one object: scaling $B$ to accomplish EC, and further scaling $B$ to accomplish PO (Fig. 10).
(c) Scaling both objects: scaling $B$ to get EC, then scaling $A$ to get PO (Fig. 11).

The complete neighborhood graph for smooth transitions in relations (due to movement of objects via translation, rotation, and scaling) is given in Fig. 12.
The middle branch DC-EC-PO-EQ is the result of translations and no scaling. The left and right branches (DC-EC-PO-TPP-NTPP) and (DC-EC-PO-TPPc-NTPPc) are the result of translation-translation-scaling-translation transformations. More complex spatio-temporal queries may have multiple (or incomplete) answers, because some transformations may cancel each other along the way. For example

(i) DC to EC can result from translation of DC or EC, or scaling of DC or EC. So we do not know what happened from DC to EC; there is no way to tell with the current representation. We need to know additional information about the objects.

(ii) PO to EQ can result from a translation PO-EQ or scale-translate-scale-translate, PO-TPP-TPPc-EQ.

(iii) PO to TPP can result directly from PO-TPP using translation or scaling; or via PO-TPPc-NTPPc-EQ-NTPP-TPP. In this case scaling TPPc-NTPPc-EQ and EQ-NTPP-TPP may implicitly be undone to get PO-EQ-TPP.
Similarly, there can be complex spatio-temporal transformations to reach the destination relation. It is impossible to tell from the graph unless a temporal record of events is available. These are open questions for the future.

One focus of QSR is the modeling of and reasoning about \textit{changes} in topological relationships. For example, Egenhofer and Al-Taha introduced the concept of topological distance, and used it to create a conceptual neighborhood graph\textsuperscript{31} for a visual representation of the changes that transform the spatial relation between a pair of objects into another relation. A similar approach is employed in VRCC-3D+ to effectively answer questions such as “is object B a possible topological transformation of object A.”

6.1. \textit{Construction of a “smooth transition” conceptual neighborhood graph}

There are several ways to construct a connected relation graph. However, the result may not be a “smooth transition” graph. The Snapshot Model\textsuperscript{32} is independent of the smooth transition between the relations. When all minimum length edges are connected, it may turn out to be a graph which is disconnected; or if it is connected, it may not be even a tree. If it is a tree, it may not be a Kruskal or prim minimum spanning tree (MST) or single source shortest path (SSSP) (Dijkstra) tree. The snapshot model simply calculates the edge length as the sum of absolute differences from 9-intersection matrices for the relations. The Snapshot algorithm is:

\textbf{Snapshot algorithm}

\begin{verbatim}
G = (V, E)
CNG=Null;
m = min length of edges in E;
While CNG is disconnected do
  For all edges of length m, /* Connect them*/
    Insert the edges in CNG
  If CNG is disconnected, let m be next min length
EndWhile
\end{verbatim}

In reality, deformation is a continuous transformation of objects that changes the relations. In contrast to the Snapshot Model approach, the smooth transition method counts the number of deformations that may determine the change in the relation. To date, it has been applied to line-region connection\textsuperscript{32} only. In that work, the knowledge of the change process is effectively ignored, as mostly neighbors are the same in both snapshot and smooth transition methods. All the analysis is specific to the example under consideration. Unfortunately, all the discussion centers around line-region connection (i.e. the specific case of connection between a 1D line and a 2D region). It does not shed light on the case where two objects are of homogeneous dimensions. A little more general definition of space partitioning for conceptual
neighbors is given, which deals with a GIS application. Yet that again falls short of the discussion of homogenous dimension regions.

Here, we give another method to address all such queries for regions of homogeneous dimensions in 2D and 3D. A region in space can be partitioned into three parts: interior, boundary, and exterior (as shown in Fig. 13). Pairwise intersections of two objects gives a $3 \times 3$ 9-intersection matrix. So each relation can be represented by a 9-intersection matrix as shown in Table 2.

The interpretation of the intersections depends on their use, which varies and is temporal. For example, as sets, the intersections are non-empty or empty sets. For “is there an intersection or not”, the intersection is a logical value (true or false). We may represent intersections as numerical values. If we want to know how many component intersections are true or non-empty, we may use 1 for true intersection and 0 for false (empty intersection), and sum the numerical values for intersections. Unfortunately, this number does not give the distance between objects. So the 9-intersection table representation is context dependent. We can use this information for numerical calculation of distance between the relations. If $R_1$ and $R_2$ are $3 \times 3$ matrices, the matrix $\text{abs}(R_1-R_2)$ has entries with element-wise absolute values of differences. In other words, an element in $R_1-R_2$ is 0 if the corresponding entries in the two relations are identical, and 1 if they are different. For example, we can use the symbols shown in Table 14 for these absolute values where $I$ = interior, $B$ = boundary, and $E$ = exterior.

Let $D$ be the sum of all the entries, $D = II+IB+IE+BI+BB+BE+EI+EB+EE$. Since 9-intersections are equivalent to 4-intersections, we shade the corresponding $2 \times 2$ matrix entries in Table 14. For the $2 \times 2$ version, let $D$ be the sum of all the entries, $D = II+IB+BI+BB$. The value of $D$ ranges from 0 to 4. The value $D = 0$ refers to itself. It means no significant change occurred from $R_1$ to $R_2$ ($R_1 \equiv R_2$) as a result of the movement of objects.

![Fig. 13. A cube depicting the partitioning of space into three parts.](image)

| Table 14. Numerical constants associated with intersections. |
|---|---|---|
| $II$ | $IB$ | $IE$ |
| $BI$ | $BB$ | $BE$ |
| $EI$ | $EB$ | $EE$ |
The value $D = 4$ refers to non-smooth transitions such as DC to PO. We design the algorithm for $D = 1, 2, 3$ as follows. This algorithm STA _RCC (Smooth Transition Algorithm) determines whether $R_1$ and $R_2$ are a smooth transition of each other. If they are, it returns true and we can record the relation pair in the set of smooth transitions, otherwise, it returns false.

**Algorithm 1**

Given: A pair $R_1, R_2$ of relations in $B$.
Required: A logical value depending on whether $R_1$ is a smooth transition of $R_2$

Boolean STA _RCC(R1,R2) /* Smooth Transition Algorithm */

$|II, IB, BI, BB| = abs(R1-R2)$
$D = II+IB+BI+BB$
If ($D = 1$)
  If ($II=0$)
    Return true
  else
    Return false
elseIf ($D = 2$)
  If ($II = 0 \& (IB=0|BI=0|BB=0) \& (R1! = R2c)$)
    Return true
  else
    Return false
elseIf ($D = 3$)
  If ($BB = 0$)
    Return true
  else
    Return false
else
  Result = false
End Algorithm /* Smooth Transition Algorithm*/

Similarly algorithm STA _Obs can be written for the VRCC-3D+ obscuration relations (which are characterized by a 3-intersection and qualitative depth parameter). As in the case of the integrated composition table discussed in Sec. 4.4, we can integrate the RCC and Obs parts similarly to design and implement a comprehensive smooth transitions algorithm.

**Algorithm 2**

Given: a set $S$ of relations in $B$.
Required: Find CNG, conceptual smooth transition neighborhood graph

/*
Initialize conceptual neighborhood graph CNG to empty list of all conceptual neighbors
CNG = null

/*
   Use STA_RCC to determine which pairs are conceptual neighbors
   Retain all conceptual neighbors
*/

For \((R_1, R_2)\) in \(S\)
   If STA_RCC\((R_1, R_2)\)
      add \((R_1, R_2)\) to CNG
EndFor
End Algorithm

Algorithm 3
Given: A relation \(R\) in \(B\).
Required: Find \(\text{CNR}\), the set of conceptual neighbors of \(R\)
/*
   Create a set of all conceptual neighbor pairs, CNG.
   Initialize \(\text{CNR}\) to empty list of conceptual neighbors of \(R\)
*/

CNR = null;

/*
   Use CNG to determine the conceptual neighbors of \(R\)
   Retain all conceptual neighbors
*/

For \((R_1, R_2)\) in CNG
   If \((R == R_1)\) add \(R_2\) to \(\text{CNR}\)
   ElseIf \((R == R_2)\) add \(R_1\) to \(\text{CNR}\)
EndFor
End Algorithm

Algorithm 4
Given: Relations \(R_1, R_2\) in \(B\).
Required: Find a path \(P\) of conceptual neighbors from \(R_1\) to \(R_2\).
/* Use Smooth transition algorithm STA_RCC to find conceptual neighborhood CNG.
Start with \(R_1\) and find conceptual neighbors using depth/breadth first search until \(R_2\) is found. Start with \(T\)
*/

\(T\) is the set of all relation under consideration
\(S = \{R1\}\)
\(T = T\setminus\{R1\}\)
Repeat
   /* Find \(s\) in \(S\) and \(R\) in \(T\) that are neighbors */
   if \(((s, R)\) in CNG)
Algorithm 4 determines a minimal transitions path $P$ of conceptual neighbors from $R_1$ to $R_2$. These algorithms can be used to answer other queries such as what is the next move from any position on a path from $R_1$ to $R_2$.
These algorithms use a 4-intersection framework. With a 4 or 8-intersection model, the edge length may vary, but the smooth transition graph does not. Using 9-intersection and 4-intersection models, the smooth transition graphs are identical. In this case, the conceptual neighborhood graph is always unique as shown in Fig. 14. Different colors indicate different snapshot distances in both models. All edges of the same snapshot length are of the same color. It shows that 9-intersection model edges are of 4 different lengths: 1, 3, 4, 6. In contrast, in the 4-intersection model, edges are of three different lengths: 1, 2, 3.

The composition table shown earlier in Table 6 can be visually represented by means of the conceptual neighborhood graph structure in Table 15. The neighborhood pattern is adapted from Rodriguez²⁵.
A relation transition path is a directed path between two spatial relations. The path may be the conceptually shortest distance between the state at time $t_1$ and the state at time $t_2$, or some other parameter count of different relations that lead from the source relation to terminal relation. Usually the transition state can be represented graphically (relations become the nodes and the transition becomes the edge) or by means of a 2D matrix (row and column headers become the relations and matrix entries become the conceptual paths). For a large set of relations, the network structure can be lost in the matrix representation or graph display.

Topology is an important tool to explain movement behavior pattern conceptualizations by means of continuous transformations such as translation, rotation, and scaling. Several approaches have been used to integrate space and time into formal frameworks. The cognitive aspects of static spatial relations have been researched, but the movement patterns via transformations and occlusion in 3D have not received the same attention. Thus the conceptual neighborhood paths are useful for identifying spatially smooth transitions of objects. Specifically, they can be used to order the relations for predicting intermediate relations.

7. Consistency-Checking

7.1. Composition table

Once a conceptual neighborhood graph has been constructed, we want to ensure that it is complete and consistent. Checking consistency with infinite domains is generally intractable. The path consistency problem is $O(n^5)$, an efficient algorithm is of $O(n^3)$.

The composition table is used without any specific instantiation of objects. It can be applied to any configuration. From the configuration graph, a minimal subgraph can be obtained which is equivalent to the original graph. That is, the same inferences can be made from the subgraph. The minimal graph is not necessarily a tree; it is neither a minimal cost spanning tree (MST), or a single source shortest path tree (SSSP). The subgraph can be obtained by applying the logical consistency algorithm in conjunction with composition table, path consistency. Checking logical consistency becomes a constraint satisfaction problem. The properties of such a graph are

- Each node has a self-loop, indicating no significant change in the spatial movement of the objects.
- Each edge is bidirected because each relation has a converse.

Most of these calculi can be formalized as abstract relation algebras, such that reasoning can be carried out at a symbolic relation level rather than an object level. For computing solutions of a constraint network, the path-consistency algorithm is an important tool.
It is obvious in temporal interval algebra that for any three times \( t_1, t_2, \) and \( t_3 \), if \( t_1 < t_2, t_2 < t_3 \) then the relation \( < \) (e.g. \( t_3 < t_1 \)) is not consistent because \( "<" \) is irreflexive. In our spatial domain, for objects \( A, B, \) and \( C \), if \( PP(A, B), PP(B, C) \), then the relation \( PP \) (i.e. \( PP(C, A) \)) is inconsistent because \( PP \) is irreflexive. In general, as the amount of information about the objects of interest increases, so too does the chance of inconsistency in the defined relations.

The composition table is used to refine relations: a relation \( R \) is a refinement of a relation \( S \), if \( R \subset S \). If we do not know the relation between \( A \) and \( B \), or we are uncertain about a relation between \( A \) and \( B \), but we know the relation of \( A \) and \( B \) to a third object \( C \), we can refine uncertainty by using a composition table. More precisely, if \( R_{ij} \) is a possible set of relations between two objects \( A_i \) and \( A_j \), we can replace \( R_{ij} \) with \( R_{ij} \cap (R_{ik}o_k R_{kj}) \) for all object instantiations \( A_k \). If the final \( R_{ij} \) is empty, the relation is inconsistent; otherwise, it is consistent in the current state.

If the relation between two objects is considered as a constraint, then the logical constraint satisfaction problem is formulated as consistency-checking, or path consistency. If we know the relation between \( A \) and \( B \), we can extend/infer the path consistency relation from \( A \) and \( B \) to \( C \). To do so more precisely, if \( R_{ij} \) is a possible set of relations between \( A_i \) and \( A_j \), we apply the following criteria recursively to object instantiation \( A_k \) until a failure (inconsistency) or convergence is reached.

\[
R_{ik} \leftarrow R_{ik} \cap (R_{ij}o_k R_{jk}) \\
R_{ik} \leftarrow R_{ik} \cap (R_{ij}o_k R_{ik})
\]

There can be several paths between two nodes. However, it is sufficient to consider paths of length 2 connecting relations nodes,\(^{35}\) that is, \( R_{ij} = \cap_{k=1,n} (R_{ik}o_k R_{kj}) \) for \( 1 \leq i, j \leq n \). A relation between two nodes is completely known if it is a unique true intersection. If it is empty, then the relation is inconsistent. If it is singleton, it is called a Crisp relation; otherwise, it is an Ambiguous (vague, uncertain) relation. If the initial graph is consistent and complete, the minimum subgraph must be unique. The uniqueness algorithm is exhaustive using composition table.\(^{24}\)

For some applications it is useful to pre-determine the consistency of data facts, known as entailment-checking. In others, it may be necessary to check the correctness of logical inferences, which is known as relation edge consistency, or path consistency. Looking up information in a composition table can greatly simplify the process of consistency checking. The composition table is useful for both entailment checking and path consistency checking to ensure the correctness of all logical inferences.

An enormous amount of knowledge can be predicted from a small set of binary relations. In general, certain disjunctive relations are formed to check consistency and to extract complete information. If all the knowledge is stored using zero order logic, then it is computationally unsatisfactory, NP-complete, except for simple cases. Alternately, the problem can be formulated as a CSP problem. First order logic provides a solution when the domain is infinite. It has two problems: completeness and tractability. Efficient solutions have been found for many classes of CSP.\(^{37}\)
7.2. Triangle consistency

Since the composition table is used for efficient implementation of inferences, does it provide all inferences needed for complete reasoning? If not, under what conditions does it provide a complete test of consistency?

The most common facts for spatio-temporal consistency are three relations instantiated with three objects. This is referred to as triangle consistency or 3-consistency in constraint satisfaction problems (CSPs). In spatio-temporal reasoning, the three relations are not exclusively base relations. Each disjunctive relation is instantiated with base relations for a triangle consisting of specific relation values. For the composition table to be useful, it is necessary to determine the conditions under which 3-consistency is complete. In general, the triangle-consistency does not provide the complete consistency. For example, a circle can be externally connected to a maximum of six other circles; this constraint is not enforceable in general by triangle checking. A similar example can be constructed for non-convex objects.

8. Spatio-Temporal Considerations Spatio-Temporal Movements

Contrary to mathematical or physical theories about space and time, qualitative constraint calculi provide efficient reasoning about objects located in space and time. For this reason, the limited expressiveness of qualitative representation formalism calculi is computationally efficient if such reasoning tasks are integrated in applications. For example, some of these calculi may be used for navigating and communicating with a mobile robot.

If the user has created two or more states that represent arrangements of the objects at different abstract time periods, the VRCC-3D+ system can be asked to determine if state $t_j$ is one deformation of objects from state $t_i$ where $i < j$. This is accomplished by finding all object pairs that are one deformation from their current relation. We can use the conceptual neighborhood graph to determine if there is a path of length one in the neighborhood graph from the relation at $t_i$ to the relation at $t_j$. After finding all object pairs with such a path, we determine if that change creates a spatial arrangement that is equivalent to the arrangement at time $t_i$. This can easily be done by making that change, and using the All-Pairs-Relation-Detection Algorithm to recalculate the relations that could have changed. The Dynamic-Path-Consistency Algorithm can then be used to find inconsistencies. After the Dynamic-Path-Consistency Algorithm is completed, we compare the resulting arrangement to state $t_j$; if they match, state $t_j$ is a deformation of state $t_i$.

If the two arrangements (resulting from application of the Dynamic-Path-Consistency Algorithm) are not the same, the next pair is chosen and the same process is applied. If no pair of relations is found that results in a match with $t_j$, the system will ask the user to create another state $t_x$, where $t_i < t_x < t_j$, or the system can be asked to generate a new possible intermediate state $t_x$. The user may pick from the system-generated possible states that are created by interpolating the 4-intersection for 3D
and 3-intersection for perspective projection and the InFront attribute value of the configurations at times $t_i$ and $t_j$. This amounts to discovering new states (and hence knowledge) interactively.

In general, spatio-temporal programming facilitates the specification of spatial relationships that must hold at particular time periods. Related work includes spatio-temporal reasoning, spatio-temporal constraint logic programming and spatio-temporal simulation. VRCC-3D+ would best be categorized as spatio-temporal visual constraint logic programming. However, VRCC-3D+ surpasses the capabilities of many spatio-temporal programming languages; namely, it allows the user to not only specify, but also to discover, spatio-temporal knowledge.

9. Future Work

In addition to the algorithmic (implementation) efficiency considerations, clearly there is a need for a VRCC-3D+ graphical user interface that would allow the user to interactively view and manipulate the objects and the corresponding spatial relationships. We plan on developing a visual programming environment that facilitates spatio-temporal representation of and reasoning over a collection of 3D objects. This system effectively will allow the user to create visual “programs”, utilizing the region connection calculus to identify and enforce the spatial constraints that logically must hold between the objects over a series of abstract time periods.

10. Conclusion

This work represents a substantial advance over the previous work in QSR. We showed that the 4-intersection model is equivalent to the 9-intersection model. As a result, all the eight RCC-8 spatial relations can be represented with 4-intersection rather than 9-intersections. Using the 4-intersection model saves 60% of the computational effort, an important consideration when working with 3D (not just 2D) data.

We presented a divide-and-conquer algorithm for constructing large composition tables. First we showed that an $8 \times 8$ composition table can be obtained from smaller, $4 \times 4$, $2 \times 2$, even $1 \times 1$ tables. Then we applied the same criteria to construct a $37 \times 37$ composition table using an $8 \times 8$ RCC-8 table (computed in 3D) and a $12 \times 12$ table in the 2D view plane, including qualitative distance from the viewer, for the VRCC-3D+ system.

In addition to using a composition table, conceptual neighborhoods play an important role in QSR. We presented a new algorithm for “smooth transition” between spatial transitions and path consistency. This is useful for spatio-temporal applications that require determination of the transformation that occurred in the pair of regions for relation $R_1$ at time $t_1$ and the relation $R_2$ at time $t_2$, or what can be expected of relation $R_2$ at a future time $t_2$.

This paper addressed the evolution of RCC from axiomatic and intersection models, efficiency of the intersection computations, computation of large composition
tables, detection of consistency, refining uncertainty, identifying smooth transition of spatial relations, and extending 3D spatial relations to include occlusion support. All these advanced algorithms have been applied to VRCC-3D+, which supports not only relations in 3D, but also obscuration in 3D relative to a viewer. Our approach is intuitive, adaptable, and efficient. We believe that the methods presented herein will provide useful guidelines for other studies in automated spatial reasoning.

References


22. A. G. Cohn, B. Bennet, J. Dooday and N. M. Gotts, Qualitative spatial representation and reasoning with the region connection calculus, *GeoInformatica* 1(1) (1997) 1–44.


27. T. Mossakowski and R. Moratz, Qualitative reasoning about relative direction on adjustable levels of granularity, arXiv, 1011.0098.


