Visual Spatio-Temporal Programming with a 3D Region Connection Calculus

Julia Albath, Jennifer L. Leopold, Chaman L. Sabharwal
Department of Computer Science
Missouri University of Science and Technology
Rolla, MO, USA
jgadkc@mst.edu, leopoldj@mst.edu, chaman@mst.edu

Abstract—The ability to perform qualitative spatial reasoning over a collection of 3D objects would be useful for a variety of problem domains, including biomedical analyses, geographic information systems, and mechanical modeling. In addition to the logical consistency checking that would be required for such a system, clearly there would be a need for a graphical interface that would allow the user to view and manipulate the objects and the corresponding spatial relationships. Additionally, consideration should be given to representing how those spatial relations can change over time. Herein we present a visual programming environment that facilitates spatio-temporal representation of and reasoning over a collection of 3D objects. This system effectively allows the user to create visual “programs”, utilizing a region connection calculus to identify and enforce the spatial constraints that logically must hold between the objects over a series of abstract time periods.

Keywords—qualitative spatial reasoning; constraint logic programming; region connection calculus

I. INTRODUCTION

Spatial reasoning is an important human skill that is employed in many tasks such as biomedical diagnostics, architectural and mechanical design. In such applications, a collection of 3D objects may be available for the analysis, possibly organized as a time series, wherein the spatial changes that occur from one time period to the next are of particular interest.

Automated spatial reasoning systems typically use logical consistency checking to detect inconsistencies and enforce constraints. Given the graphical nature of the data, clearly a visual user interface to such a reasoning system would be beneficial. To a certain degree, CAD/CAM applications have addressed this need. However, the reasoning capabilities of such systems are very limited; there is no underlying support for knowledge discovery, and the constraints that are enforced are often domain-specific.

Herein we introduce a visual programming environment, VRCC-3D, which allows the user to create visual “programs” that represent a collection of 3D objects and the (generalized) spatial relations that hold between the objects. A region connection calculus, RCC-3D, is utilized to identify and enforce spatial constraints that logically must hold between the objects over a series of successive time periods.

As shown in Fig. 1, the VRCC-3D user interface has two panes: on the left hand side is a graphical representation of a collection of 3D objects, and on the right hand side is a textual representation of the RCC-3D relations that hold between the various objects. Any changes made to the textual representation will be reflected in the graphical representation, and vice versa. The initial display can be created in two different ways. A file containing the 3D object data can be loaded, and the RCC-3D spatial relations determined [1] and displayed textually. Alternately, a (textual) file of RCC-3D relations can be loaded, and the system will display a graphical representation of a corresponding collection of objects, produced by prompting the user with a series of questions about each object (e.g., selecting from a standard set of shapes like cubes and spheres, choosing the size and orientation, etc.).

Both of these “scene initialization” modes further allow the user to specify an abstract time period (e.g., $t_1, t_2, t_3, \ldots$) to be associated with a particular set of objects and spatial relations. These different “states” can then be analyzed to determine if there exist RCC-3D relations that could hold between the objects at successive time periods, effectively supporting (or refuting) a spatial transformation of the objects from one state to another state.

![Figure 1. VRCC-3D Interface Example](image-url)
In the next section, we provide a very brief overview of related work in qualitative spatial reasoning, and visual constraint logic and spatio-temporal programming systems. We then discuss how VRCC-3D utilizes a conceptual neighborhood and a composition table to facilitate consistency checking and spatial transformation determination based on RCC-3D.

II. RELATED WORK

A. Constraint Logic Programming

Logic programs which have constraints in the clauses are known as constraint logic programs (CLP). Constraints (specified statically or generated dynamically) are tested in order to control execution of the program. A survey introduction to CLP is provided by Jaffer and Maher [10].

VRCC-3D employs a similar approach, whereby each program “state” represents a set of constraints on the spatial arrangement of the objects at an abstract time period. A number of visual logic programming systems [12], [11], [16] and visual constraint programming systems [2], and [15] have been proposed. However, no system has previously been developed to support visual constraint logic programming based on a region connection calculus.

B. Spatio-Temporal Programming

In general, spatio-temporal programming facilitates the specification of spatial relationships that must hold at particular time periods. Related work includes spatio-temporal reasoning [6], spatio-temporal constraint logic programming [13], and spatio-temporal simulation [7]. VRCC-3D would best be categorized as spatio-temporal visual constraint logic programming. However, as will be discussed later in this paper, VRCC-3D surpasses the capabilities of many spatio-temporal programming languages; namely, in allowing the user to, not only specify, but also to discover spatio-temporal knowledge.

C. Qualitative Spatial Reasoning

Qualitative Spatial Reasoning (QSR) concerns the issues related to automating human spatial analytical processes [8]. A region connection calculus was introduced in [14], and has been one of the foundational theories for QSR.

One focus of QSR is the modeling of and reasoning about 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 [5], 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.”

III. RCC-3D

Before discussing the specific functionality of VRCC-3D, we provide details about the underlying region connection calculus, RCC-3D, and two associated tools that are used to solve spatial reasoning tasks.

A. RCC-3D Overview

The fundamental axioms of RCC-3D are Parthood and Connectivity. For two distinct objects A and B, we use $A \cap B$ to represent everything common to both A and B; $A-B$ is used to denote everything that is in A, but not in B. The closure of A is $\overline{A}$, and the boundary of A is $\partial A$. $A^o$ denotes the interior of A, which is everything that is in the closure of A, but does not include the boundary of A. $A^c$ or $\overline{\overline{A}}$ is the exterior of A; it represents everything that is not in $\overline{A}$. The interior, boundary, and exterior of an object are disjoint, and their union is the universe. $P(A,B)$ is defined as A is part of B if A is a subset of B; specifically, $A \cap B = \emptyset$. A is connected to B, denoted by $C(A,B)$, if $\overline{A} \cap B \neq \emptyset$. Additionally, we define a predicate Proper Part (PP) as follows:

$$\text{PP}(A,B) \iff P(A,B) \land \neg P(B,A)$$ (1)

The eight base relations of RCC-3D (which comprise the entire set of relations for RCC-8 [17]) can be defined in terms of $P$, $C$, and $PP$ as listed below:

$$\text{PO}(A,B) \iff \exists (C \neq \emptyset) [P(C,A^o) \land P(C,B^o)]$$ (2)

$$\text{EQ}(A,B) \iff A = B$$ (3)

$$\text{EC}(A,B) \iff C(A,B) \land (A^o \cap B^o = \emptyset)$$ (4)

$$\text{DC}(A,B) \iff \neg C(A,B)$$ (5)

$$\text{TPP}(A,B) \iff \text{PP}(A,B) \land \exists C(\neq \emptyset) [C(A,B) \land \text{EC}(C,B^o)]$$ (6)

$$\text{NTPP}(A,B) \iff \text{PP}(A,B) \land \forall C [C(A,B) \land \text{EC}(C,B^o)]$$ (7)

$$\text{TPPc}(A,B) \iff \text{PP}(B,A) \land \exists C(\neq \emptyset) [C(A,B^o) \land \text{EC}(C,B)]$$ (8)

$$\text{NTPPc}(A,B) \iff \text{PP}(B,A) \land \forall C [C(A,B^o) \land \text{EC}(C,B)]$$ (9)

To facilitate automated reasoning, each of the relations must have a converse. For example $\text{TPPc}$ and $\text{NTPPc}$ are the converses of $\text{TPP}$ and $\text{NTPP}$, respectively. The converse of a relation is denoted with a $c$ at the end of the relation name. However, this convention is not necessary for the symmetric relations $\text{PO}(A,B)$, $\text{EQ}(A,B)$, $\text{EC}(A,B)$, and $\text{DC}(A,B)$.

For some types of spatial reasoning it is necessary to consider the obscuration that can occur when the objects are seen through orthographic projection on any of the principal planes in $\mathbb{R}^3$. When considering that object A obscures object B, it is implied that object A is closer than object B to the perspective reference point. For our discussion, we assume that the direction of the projection (i.e. the line of sight) is orthogonal to the plane of projection, and that the plane of projection is one of the principal planes (xy, yz, or zx). The obscuration of objects is determined by projecting the objects on a plane P.
Unlike other regional connection calculi such as RCC-8, RCC-3D differentiates between complete and partial obscuration. Partial obscuration is indicated when the relation’s name includes a $p$ at the end. For example, $ECP(A,B)$ is true if objects $A$ and $B$ are externally connected and object $A$ partially obscures object $B$: $ECP(A,B)$ is true if objects $A$ and $B$ externally connect, and object $A$ completely obscures object $B$. The additional relations of RCC-3D (which consider obscuration) are given below, where $A_p$ and $B_p$ represent the respective projections of $A$ and $B$ on the plane $P$, and where $P$ is $xy$, $yz$, or $zx$. Note that $PO$ has now been distinguished as $PO_p$ and $PO$.

\[ \begin{align*}
POpp(A,B) &= PO(A,B) \land PO(A_p,B_p) & (10) \\
POp(A,B) &= PO(A,B) \land (TPPc(A_p,B_p)) \lor \text{NTPPc}(A_p,B_p) \quad (11) \\
EQp(A,B) &= A = B \quad (12) \\
ECp(A,B) &= EC(A,B) \land PO(A_p,B_p) \quad (13) \\
EC(A,B) &= EC(A,B) \land (TPPc(A_p,B_p)) \lor \text{NTPPc}(A_p,B_p) \quad (14) \\
DCpp(A,B) &= DC(A,B) \land PO(A_p,B_p) \quad (15) \\
DC(A,B) &= DC(A,B) \land (TPPc(A_p,B_p)) \lor \text{NTPPc}(A_p,B_p) \quad (16)
\end{align*} \]

\[ \begin{align*}
 & \text{TABLE I. 9-INTERSECTION FOR DC} \\
\begin{array}{ccc}
A' & B' & \tilde{B}' \\
\text{A} & \varnothing & \varnothing & \varnothing \\
\tilde{A} & \varnothing & \varnothing & \varnothing \\
A' & \varnothing & \varnothing & \varnothing
\end{array}
\]

The conceptual neighborhood graph is created by considering the closest topological distance (as calculated for each pair of relations), as well as possible deformations of scaling, translation, and rotation. The neighborhood of an object is an equivalence class of all objects that are topologically transformable into each other. In particular, it is the equivalence class of all objects that are topologically transformable into it with the least number of transformations.

Expanding on this approach for RCC-3D, we combined the 9-Intersection for the RCC-8 base relations with a 4-Intersection for the projections onto the specified plane. The 4-Intersection considers the intersections of object $A$’s interior and exterior with object B’s interior and exterior; the boundaries are not relevant in this context. A $2 \times 2$ matrix, $M$, is used to represent these intersections as shown in Table III.

\[ \begin{align*}
 & \text{TABLE III. MATRIX M} \\
\begin{array}{ccc}
A' & B' & \tilde{B}' \\
A' & \varnothing & \varnothing & \varnothing \\
\text{A} & \varnothing & \varnothing & \varnothing
\end{array}
\]

There are 16 combinations for $M$ using the values of empty and not-empty. Of those 16 combinations, only five are possible considering reality constraints; for example, unless one of the objects is the universe (in which case it has no exterior), the intersection of the exteriors cannot be empty. These match the five relations of the RCC-5 calculus [14]. The five relations and their matrix representation are as follows, each specified as a 4-tuple of intersections ($A \cap B$, $A \cap B'$, $A' \cap B'$, $A' \cap B$): $DR=(\varnothing, \varnothing, \varnothing, \varnothing)$, $PO=(\varnothing, \varnothing, \varnothing, \varnothing)$, $EQ=(\varnothing, \varnothing, \varnothing, \varnothing)$, $PP=(\varnothing, \varnothing, \varnothing, \varnothing)$, and $PPc=(\varnothing, \varnothing, \varnothing, \varnothing)$.

$DR$ represents the situation where no obscuration occurs between the two objects in a particular plane. $PO$ is indicative of partial obscuration. $EQ$, $PP$, and $PPc$ represent the cases of full obscuration. The results of applying these intersections to the RCC-3D relations (in terms of the 9- and 4-Intersections, where applicable) are shown in Table IV.

\[ \begin{align*}
 & \text{TABLE IV. 9-INTERSECTION FOR TPP} \\
\begin{array}{ccc}
A' & B' & \tilde{B}' \\
A' & \varnothing & \varnothing & \varnothing \\
\tilde{A} & \varnothing & \varnothing & \varnothing \\
A' & \varnothing & \varnothing & \varnothing
\end{array}
\]
The 9-Intersection can be used to determine inter-relation topological distance, and the 4-Intersection can be used for the determination of the intra-relation topological distance. Here intra-relation means within the same (base) relation (e.g., DC_{op} to DC_{p}, or from EC_{p} to EC_{lp}), whereas inter-relation means between (base) relations (e.g., from EC_{op} to DC_{op}, or PO_{p} to TPP_{c}). It is important to distinguish between these two types of distances. A total topological distance of (9-Intersection + 4-Intersection = Total) 0+2=2 is different from a distance of 2+0=2. This also is significant when we consider that a change in the 9-Intersection often implies a change in the 4-Intersection.

The computed topological distances for the RCC-3D relations are shown in Table V, where the first number is the topological distance in the 9-Intersection and, where applicable, the second number (y in the expression x+y) is the topological distance in the 4-Intersection.

The entries in Table V are important for building the RCC-3D conceptual neighborhood graph. The inter-relation distance between two base relations is the number of entries that differ between their 9-Intersection representations. For determining obscuration, objects are projected on a plane and the relation uses the additional 4-Intersection representation. The intra-relation distance between two relations is then the number of entries that differ between their 4-Intersection representations. The conceptual graph uses both inter-relation and intra-relation distances. In general, the distance of a relation from another relation R is represented by x_{inter} + y_{intra}. If x_{inter} = 0 and y_{intra} = 0, then the neighbor is the relation itself. If x_{inter} = 0 and y_{intra} ≠ 0, then the neighbor is from the same base relation, e.g., DC and DC_{p}, PO_{p} and PO_{sp} are neighbors. If x_{inter} ≠ 0, then the neighbors are from different base relations, e.g. PO_{p} and EC_{p} are neighbors. To determine conceptual neighbors of a relation, it is first necessary to find those relations which have minimum distance among inter-relation distances, \( x = \min(x_{inter}) \). Once this is done, from among these relations we must find those relations that have minimum distance among intra-relation distances. The closest nodes are those which have distance min \( \{ x + y_{intra} \} \). Here we choose the relations where first, the 9-Intersection is smallest, and then the 4-Intersection is smallest. For example, in Table V the relation PO_{p} has neighbors of different base relations as the relations EC, EC_{lp}, EC_{p}, TPP, and TPP_{pc} with minimum inter-relation distance 3. Among these, only EC_{p} and TPP_{pc} have distance 3+0; others are at distance x+y, x>2, y>0. So EC_{p} and TPP_{pc} are the only closest neighbors of PO_{p} in the conceptual neighborhood graph.

The complete conceptual neighborhood graph for RCC-3D is depicted in Fig. 2.

IV. VRCC-3D

The composition table and the conceptual neighborhood play an important role in the VRCC-3D system’s consistency checking and determination of spatial transformations across abstract time periods. This functionality can be illustrated with two possible use cases.

A. An Example of Inconsistency Checking

When changes are made to the graphical representation of the collection of objects (in the left hand pane of the VRCC-3D interface), the system must update the corresponding textual display of RCC-3D relations in the right hand pane. Because the change was made directly to the graphical representation, we know that the resulting spatial arrangement of the objects (and associated RCC-3D relations) will be consistent. Updates to the matrix of RCC-3D relations will take place in both the entire column and row associated with the object that has been altered. Specifically, the All-Pairs-Relation-Detection Algorithm [1] is applied to recalculate the affected relations; this algorithm uses the composition table to narrow down the possible relations that could hold, in the process of determining the single relation that does hold between a pair of objects.

A simple example of this use case is illustrated in Fig. 3, wherein the user has altered the scene shown earlier in Fig. 1 by moving object A. Clearly this requires recalculation of the RCC-3D relations in row A and column A (which are highlighted in the RCC-3D relation matrix that also is displayed in Fig. 3). The relation for (A,B) changes from PO_{sp}(A,B) to DC_{op}(A,B). The relations for (A,C) and (A,D) do not change in this particular case, but still are recalculated; in future work, we will investigate ways to detect and avoid such unnecessary efforts.
Suppose that the user instead directly updates entries in the RCC-3D relation matrix (in the right hand pane of the VRCC-3D interface). All object pairs which have been changed are then added to an “updated list.” We use the Dynamic Path Consistency Algorithm [1], which in turn uses the composition table, to check whether those changes have introduced any inconsistencies in the associated spatial relations.

If any inconsistencies are detected, the VRCC-3D system displays an error message, and gives the user an opportunity to make corrections. In no inconsistencies are found, the graphical display needs to be updated; an efficient implementation of this particular functionality is currently under development.

B. An Example of Constraint Determination over Time

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 \( t_i \) is one deformation removed from state \( t_j \) where \( i > j \). This is accomplished by finding all object pairs that are one deformation from their current relation. We 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 \( t_i \); if they match, state \( t_i \) is a deformation of state \( t_i \). For example, suppose that the configuration of objects shown in Fig. 1 has been designated as state 1, and the arrangement shown in Fig. 3 represents state 2. The image in Fig. 4 shows the VRCC-3D user interface after the two states have been compared, and found to be equivalent under deformation.

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 \( t_i \), the system will ask the user to create another state \( t_x \), where \( t_j < t_x < t_i \) 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 9-Intersection and 4-Intersection of the configurations at times \( t_j \) and \( t_i \). This amounts to discovering new states (and hence knowledge) interactively.

V. Future Work

As previously stated, our future work will focus on efficiently updating the graphical representation of the collection of objects when a change has been made to the textual display of spatial relations. Another area of improvement will be in detecting and avoiding recalculation of relations that ultimately will not change. We hope to accomplish the latter by keeping track of the old and new locations of objects, and using quantitative methods to quickly calculate if the move necessitates a recalculation of the spatial relations; similarly, the effects of resizing of objects will be considered.

VI. Summary

In this paper we introduced VRCC-3D, a system which utilizes a region connection calculus to check the consistency of spatial relations between 3D objects. By providing a visual interface to this system, the user is able to make changes to both the graphical representation of a collection of objects as well as the textual representation of the associated spatial relations. Because the spatial changes that hold from one time period to the next may be of particular interest, we briefly discussed the roles of a composition table and a conceptual neighborhood graph to check the consistency between “states”, as well as to suggest what changes might have occurred in intermediate states. It is hoped that this work will provide a foundation for the many problem domains that involve spatial design and analysis.

TABLE IV. THE 9 + 4-INTERSECTIONS FOR RCC-3D
### Table V. The Topological Distances for the RCC-3D Calculus

<table>
<thead>
<tr>
<th>A^+ \cap B^+</th>
<th>A^0 \cap B^0</th>
<th>A^1 \cap B^1</th>
<th>d(A \cap B)</th>
<th>\delta(A \cap B)</th>
<th>A^+ \cap B^+</th>
<th>A^0 \cap B^0</th>
<th>A^1 \cap B^1</th>
<th>A^0 \cap B^0 \times</th>
<th>A^+ \cap B^+ \times</th>
<th>A^0 \cap B^0 \times</th>
<th>A^+ \cap B^+ \times</th>
</tr>
</thead>
<tbody>
<tr>
<td>DC</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DC_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DC_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EC</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EC_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PO_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>PO_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TPP_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>TPP_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EQ</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EQ_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NTPP_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>NTTP_p</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### References


