Smooth Transition Neighborhood Graphs
For 3D Spatial Relations

Chaman L. Sabharwal
Computer Science Department
Missouri University of S&T
Rolla, MO – 65409, USA
chaman@mst.edu

Jennifer L. Leopold
Computer Science Department
Missouri University of S&T
Rolla, MO – 65409, USA
leopoldj@mst.edu

Abstract. Distance between two relations can be defined by using some metric based on the qualitative or quantitative representation of the relations [1]. However, qualitative distances cannot be expressed by conventional measures. Most differentiating measures are derived from observation and experience in an ad hoc manner. The outcomes are cognitively acceptable only if they match the user’s concept of distance. We have designed an algorithm based on heuristics to derive a conceptual neighborhood supporting smooth transitions between the relations. Herein we present the results of applying the algorithm to the well-known region connection calculus, RCC-8, and to visual VRCC-3D+ that considers both 3D connectivity and obscuration.

Keywords – region connection calculus; conceptual neighborhood; conceptual distance; smooth transition; smooth transition graph.

I. INTRODUCTION

Existing methods for reasoning fall in two categories: quantitative and qualitative. Quantitative methods are useful when exact measurements are readily available. In the case of availability of partial information or in some cases to bypass quantitative information, qualitative information is more valuable to the users. Qualitative information is domain-specific which humans use to differentiate and produce results of greater practical value. For example, suppose a vehicle is approaching your car rapidly from behind, this is sufficient information for you to take precautionary action, against possible accident, without knowing the exact speed at which the vehicle distance is decreasing. As another example, think of pouring rain and the steadily rising water level of a lake by your house, it is a sufficient signal to evacuate to avoid possible flooding without knowing the exact rate of water level change, or the time the lake might flood. This paper is about representation of the distance between relations: qualitative or quantitative. Whatever form relations are represented, we design an algorithm in this paper define the qualitative distance and compare its validity with quantitative distances used in the past.

An important part of modeling spatio-temporal information and determining conceptual neighbors is that they form a conceptual neighborhood graph. Humans conceptualize moving objects as smooth transitions in the relations between objects from time t1 to time t2. Conceptual neighborhood graphs can be determined by using the Snapshot model based on distances or the smooth transition model based on deformations [1] [2] [3]. However, heretofore no simple algorithm has been made available [4].

A Conceptual Neighborhood Graph (CNG) and topological distances can be used for several practical applications; for example, they can be used for map-matching techniques to predict the location of spatial network mobile users [4]. The distance between two relations is defined by using some metric for the presentation of the relations qualitatively or quantitatively. The results are cognitively acceptable only if they match the user’s concept of distance. Qualitative distances cannot be expressed by conventional measures. For example in Fig. 3 and Fig. 4, solid lines denote ‘smooth transitions’, whereas the dotted lines represent non-smooth transition edges. The smaller the conceptual “smooth” distance, the closer the relations. Most differentiating measures are derived from observation and experience in an ad hoc manner. We have designed a hybrid algorithm that uses heuristics to derive conceptual distances that lead to smooth transitions between the relations. Herein we present the results of applying these criteria to the well-known region connection calculus, RCC-8, a set of 2D occlusion relations, and a set of relations that considers both 3D connectivity and 2D obscuration, known as VRCC-3D+.

The paper is organized into Section II: Background, Section III: Occlusion considerations and VRCC-3D+, Section IV: Algorithms, Section V: Conclusions followed by references.

II. BACKGROUND RCC-8

Region Connection Calculus [5] is based on axioms for Parthood (∈) and Connectivity (⊇) [6]. For our discussion, a region is a simply connected closed set. If a set A is a subset of B, it is denoted by ∈(A,B) (i.e., A is part of B); specifically, this requires that A⊆B, or A∩B' = ∅, where B' is the universal complement of set B. However, if a set A is a proper subset of B, it is denoted by PP(A,B) (i.e., A is proper part of B);
specifically, this requires that $A\cap B$, or $A\neq B$ and $A\cap B = \emptyset$. A set $A$ is weakly connected to a set $B$, denoted by $\emptyset(A,B)$, if $\overline{A} \cap \overline{B} \neq \emptyset$; that is, $A \cup A^c$ is weakly connected, where $A^c$, $\overline{A}$ are the interior, exterior, and closure of $A$ respectively.

Let $\mathcal{R}$ be a set of relations on pairs of bounded convex regions. If there is exactly one relation from $\mathcal{R}$ between any pair of regions, such relations are called Jointly Exhaustive and Pairwise Disjoint (JEPD). For example, RCC-8 [5] consists of the following eight JEPD relations between pairs of bounded convex objects: $DC$ (Disconnected), $EC$ (Externally Connected), $PO$ (Proper Overlap), $EQ$ (Equal), $TPP$ (Tangential Proper Part), $TPPc$ (TPP converse), $NTPP$ (NonTangential Proper Part), and $NTPPc$ (NTPP converse). RCC-8 is a relation algebra; that is, it is associative and non-commutative. The RCC-8 base relations $\mathcal{R} = \{DC, EC, PO, EQ, TPP, TPPc, NTPP, NTPPc\}$ are formally defined in point-set topology using axioms for Parthood ($\mathcal{P}$) and Connectivity ($\mathcal{C}$) as follows:

$$
\begin{align*}
DC(A,B) & \equiv \neg \mathcal{P}(A,B) \\
EC(A,B) & \equiv \mathcal{P}(A,B) \land (A^i \cap B^i = \emptyset) \\
PO(A,B) & \equiv (A^i \cap B^i = \emptyset) \land \neg \mathcal{P}(A,B) \\
EQ(A,B) & \equiv (A^i \land B^i = A^i) \land \neg \mathcal{C}(A,B) \\
TPP(A,B) & \equiv (A^i \land B^i = A^i) \land \neg \mathcal{C}(A,B) \\
NTPP(A,B) & \equiv (A^i \land B^i = \emptyset) \land \neg \mathcal{C}(A,B) \\
NTPPc(A,B) & \equiv (A^i \land B^i = \emptyset) \land \neg \mathcal{C}(A,B)
\end{align*}
$$

The RCC-8 relations are depicted as the leaf nodes of a relation tree in Fig. 1.

The RCC-8 axiomatic relations were independently defined by using a 9-Intersection formalism [7]. As shown in Fig. 2, every bounded convex region in 3D partitions the whole space into three pairwise disjoint parts: the interior (Int), the boundary (Bnd), and the exterior (Ext).

For two spatial regions $A$ and $B$, pairwise intersections of the interior, boundary, and exterior are represented by a 9-Intersection matrix, as displayed in Table 1 where $A^i$, $A^b$, $A^e$ are the interior, boundary and exterior of a region $A$.

Since for bounded objects, intersection of exteriors is always non-empty, the 9-Intersection can be replaced with an 8-Intersection, as represented by the vector $[A^i \cap B^i, A^i \cap B^b, A^i \cap B^e, A^i \cap B^b, A^b \cap B^b, A^b \cap B^e, A^b \cap B^e, (A \cap B)]$

or a qualitative descriptive vector for arbitrary objects

$$
[\text{IntInt}, \text{IntBnd}, \text{IntExt}, \text{BndInt}, \\
\text{BndBnd}, \text{BndExt}, \text{ExtInt}, \text{ExtBnd}]
$$

For example, here IntInt represents mathematical expression $A^i \cap B^i$, IntBnd, represents $A^i \cap B^b$ and BndBnd represents $A^b \cap B^b$.

Since the logical intersection value is $T$ (true) for non-empty or $F$ (false) for empty intersection, the RCC-8 base relations are described by the 8-Intersection predicate values in Table 2.

The 9-Intersection formalism has been widely used. However, at most 4-Intersections are sufficient to guarantee the same results [8]. In Table 2, the shaded entries correspond to the 2, 3, 4-Intersections necessary to compute each relation.

The distance between two relations can be defined as the number of places where the two corresponding intersection vectors differ. Tables 3 and 4 show the pairwise distances between RCC-8 relations using the 8-Intersection and 4-Intersection models, respectively. The shaded entries I the
table 3 and 4 correspond to distances used in the construction of conceptual neighborhood graph using 8-Intersection and 4-Intersection models. Diagonal entries are zero, indicating that no tangible change/movement occurred.

<table>
<thead>
<tr>
<th>TABLE 3. 8-INTERSECTION DISTANCES TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCC-8I dc  ec  po  eq  tpp  tppe  ntpp  ntppe</td>
</tr>
<tr>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>dc  0  1  4  6  5  5  4  4</td>
</tr>
<tr>
<td>ec  1  0  3  5  4  4  5  5</td>
</tr>
<tr>
<td>po  4  3  0  6  3  3  4  4</td>
</tr>
<tr>
<td>eq  6  5  6  0  3  3  4  4</td>
</tr>
<tr>
<td>tpp 5  4  3  3  0  6  1  7</td>
</tr>
<tr>
<td>tppe 5  4  3  3  6  0  7  1</td>
</tr>
<tr>
<td>ntpp 4  5  4  4  1  7  0  6</td>
</tr>
<tr>
<td>ntppe 4  5  4  4  7  1  6  0</td>
</tr>
</tbody>
</table>

A Neighborhood Graph (NG) is a graph in which the nodes represent objects and the edges represent some spatial relation. We also can represent the relations by means of a Conceptual Neighborhood Graph (CNG). The distance between a pair of relations may be defined by some other metric, such as the number of direct transitions due to deformations of objects from one relation to the other relation. Two relations R1 and R2 are Conceptual Neighbors (CN) if: (1) R1 holds for a pair of objects A, B at time t1, (2) A, B change to A’, B’ over time, where R2 is the relation between A’, B’ at time t2, and (3) no other relation has occurred between the two objects during time t1 to t2.

In Fig. 3 and Fig. 4, solid lines denote “smooth transitions’, whereas the dotted lines represent non-smooth transition edges. Note, for example in Fig. 3, the topological 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 NTPPe is smaller, 2, those two relations are not neighbors using the same criteria. In Fig. 4, the distances are calculated using the 4-Intersection. Now the same edges (PO, EQ), (TPP, TPPc), (NTPP, NTPPe), are of length 2, but PO and EQ are conceptual neighbors, and TPP and TPPc, NTPP and NTPPe still are not conceptual neighbors. In general, conceptually R and its converse, Rc, are not smooth transitions.

Connecting edges obtained by using the Snapshot model may not lead to a “smooth” transition graph as edge (PO, EQ) in Fig 3 will never occur. If we remove all the conceptually impossible edges, the remaining graph is a graph of smooth transitions. In order to accomplish this, we take advantage of the completely connected neighborhood graph and common sense heuristics to derive a “smooth” transition graph. The shaded cells in Tables 3-4 pertain to smooth transitions between the relations. The graphs corresponding to the shaded cells in the referenced tables are derived by using Algorithm 1 (RCC-8), which will subsequently be discussed in Section IV; those graphs are explicitly shown in Fig. 3-4.

Using the 8-Intersection and 4-Intersection models, the smooth transition graphs are identical. In general, Conceptual Neighborhood Graphs are not always unique. In the graphs shown in Figs. 3-4 different colors indicate different snapshot type distances in both figures. All edges of the same length are of the same color. In the 8-intersection model, the edges are of 4 different sizes, whereas in the 4-intersection model edges are of three different sizes. On comparing the graphs for 8-
Intersection and 4-Intersection, it is clear that it is easy to design an algorithm using the 4-Intersection model because the matrix entry values are closer to each other.

But the problem of determining conceptual neighbors becomes more complex as the region connection calculus becomes more expressive. For example, RCC-8 in 2D and/or 3D is insufficient for complete analysis in determining 3D relations coupled with occlusion relations. In VRCC-3D+ [9] Sabharwal et al. 2011, the authors have built an occlusion layer on top of RCC-8 (computed in 3D). Qualitative distance relative to the objects as seen by the viewer is used in addition to intersections of projections in 2D. This distance attribute value is utilized to determine occlusion of the objects A and B as seen by a viewer.

III. OCCLUSION CONSIDERATIONS AND VRCC-3D+

In VRCC-3D+ occlusion analysis is performed in two steps. First, the spatial relation is computed between the projections in 2D and then the spatial qualitative distance between the objects from the viewer is determined in 3D. The 2D analysis for projections of objects is conceptually similar to RCC-8, but the implementation is quite different. The projections alone are not sufficient to determine obscuration. As shown in Fig. 5, when B is projected on the projection plane from different locations, it has the same projection. In order to determine whether A is in front of B or B is in front of A, it is necessary to know the relative distance from the viewer.

![Figure 5. Three objects and their projection for one type of occlusion.](image)

The VRCC-3D+ occlusion relations are nObs (no obscuration), pObs (partial obscuration), eObs (equal Obscuration), and cObs (complete proper obscuration). As seen by the viewer, the value InFront(A,B) accounts for the depth of A relative to B. Hence we use a 3-Intersection and InFront to determine the obscuration uniquely: the 4-vector is (IntInt, IntBnd, BndInt, InFront). Here, it may seem the nObs-pObs-eObs-cObs relations are the same as DC, PO, EQ, PPC of RCC-8, but it is not so because for obscuration (1) these DC, PO, EQ, PPC are overloaded with 3-intersection rather than 4-Intersection (see the shaded entries in Table 5) and (2) the fourth attribute is InFront. For example, in addition to the InFront attribute, pObs is in fact a combination of PO and PP, excluding EQ, as PO(A_p,B_p)\text{PP}(A_p,B_p) accounts for the projection of A and B on a perspective projection plane for obscuration type detection. Thus, coupled with InFront, it becomes

\[
pObs(A,B) = (PO(A_p,B_p)\text{PP}(A_p,B_p))\text{InFront}(A,B).
\]

That is, we do not need to check for the boundary-boundary intersection, as seen in Table 5.

| A\begin{symbol}B | A\end{symbol}B | A\begin{symbol}B | A\end{symbol}B |
|------------------|------------------|------------------|
| A\begin{symbol}B | A\end{symbol}B | A\begin{symbol}B | A\end{symbol}B |
| A\begin{symbol}B | A\end{symbol}B | A\begin{symbol}B | A\end{symbol}B |
| A\begin{symbol}B | A\end{symbol}B | A\begin{symbol}B | A\end{symbol}B |

As in the use of the 8-Intersection or 4-Intersection vector (IntInt, IntBnd, BndInt, BndBnd) for spatial relations, here we use a 4-vector (IntInt, IntBnd, BndInt, InFront) consisting of a 3-Intersection and InFront attributes to characterize the obscuration relations. Note that for this model, BndBnd plays no part.

Since DC and EC are overloaded with the 3-Intersection, DR=DC=EC for occlusion, and nObs is characterized as:

\[
nObs(A,B) = (DR(A_p,B_p))\text{InFront}(A,B)
\]

The distance of an object from the viewer is the distance of a point on the object that is closest to the viewer. Thus the InFront attribute can have exactly the three jointly exhaustive mutually exclusive (JEPO) values. To be formally defined for depth, the converse of InFront(A,B) is denoted by InFront_c(A,B). Similar to spatial location equality EQ(A,B), the occlusion distance relation equality is denoted by InFront_e, meaning A and B are at the same distance from the viewer.

We define these obscuration relations axiomatically as follows. The occlusion relations are: nObs, pObs, eObs, cObs and InFront. So these are formally described by axiomatic predicates.

There are three types of nObs occlusion relations:

\[
nObs(A,B) = DR(A_p,B_p)\text{InFront}(A,B)
\]

There are five types of pObs occlusion relations:

\[
pObs(A,B) = PO(A_p,B_p)\text{InFront}(A,B)
\]

There are three types of eObs occlusion relations:

\[
eObs(A,B) = EQ(A_p,B_p)\text{InFront}(A,B)
\]

There are four types of cObs occlusion relations:

\[
cObs(A,B) = PP(A_p,B_p)\text{InFront}(A,B)
\]

Thus we have 15 obscuration relations corresponding to 4 categories (3 for nObs, 5 for pObs, 3 for eObs, 4 for cObs); they are listed in Table 6 using an intersection framework. The set of occlusion relations is denoted by Obs15. In Table 6 the column header for the InFront attribute has three qualitative
values Y, N, E representing “A is in front of B”, “B is in front of A”, “A is equidistant to B”, respectively.

Here the calculation of distance between two occlusion relations is different from that of RCC-8 relations. In Table 6, if we replace T & Y with 1, F & N with 0, and E with 0.5, we get Table 7. Thus IntInt, IntBnd, BndInt, InFront are overloaded with predicate values as well as numeric values as in the tables 6 and 7.

TABLE 6. OCCLUSION INTERSECTION TABLE AND INFRONT ATTRIBUTES.

<table>
<thead>
<tr>
<th>Obs-15</th>
<th>IntInt</th>
<th>IntBnd</th>
<th>BndInt</th>
<th>InFront</th>
</tr>
</thead>
<tbody>
<tr>
<td>nObs</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Y</td>
</tr>
<tr>
<td>nObs+c</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>N</td>
</tr>
<tr>
<td>nObs+e</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>Y</td>
</tr>
<tr>
<td>pObs1</td>
<td>T</td>
<td>F</td>
<td>T</td>
<td>Y</td>
</tr>
<tr>
<td>pObs2</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>Y</td>
</tr>
<tr>
<td>pObs+c1</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>N</td>
</tr>
<tr>
<td>pObs+c2</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>N</td>
</tr>
<tr>
<td>pObs+e</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>E</td>
</tr>
<tr>
<td>cObs1</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>Y</td>
</tr>
<tr>
<td>cObs+e</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>E</td>
</tr>
<tr>
<td>cObs</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>Y</td>
</tr>
<tr>
<td>cObs+c1</td>
<td>T</td>
<td>T</td>
<td>T</td>
<td>N</td>
</tr>
<tr>
<td>cObs+c2</td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>E</td>
</tr>
</tbody>
</table>

TABLE 7. THE OCCLUSION INTERSECTION TABLE MODIFIED AND OVERLOADED WITH NUMERIC VALUES

<table>
<thead>
<tr>
<th>Obs-15</th>
<th>IntInt</th>
<th>IntBnd</th>
<th>BndInt</th>
<th>InFront</th>
</tr>
</thead>
<tbody>
<tr>
<td>nObs</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>nObs+c</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>nObs+e</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>pObs1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>pObs2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pObs+c1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>pObs+c2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>pObs+e</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cObs1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cObs+e</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cObs</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cObs+c1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>cObs+c2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Now the distance between two relations can be expressed as the sum of absolute values of the attributes differences of the components in the referenced vectors. Using these distance values between the relations as distance between 4-vectors, the distance Table 8 shows the complete set of distances.

Every calculus with JEPD relations has a conceptual neighborhood [10]. Using the smooth transition criteria given by Algorithm 2 (Occlusion) in Section IV, the conceptual neighborhood graph becomes that shown in Fig. 6.

The distance table for VRCC-3D+ is 46x46. This table takes 10 pages to display, and hence is not shown here. The conceptual neighborhood graph was computed by using Algorithm 3 (VRCC-3D+), which is discussed in the next section. It has 46 nodes and 177 smooth transition edges.

A. Combining RCC-8 3D and Obscuration into VRCC-3D+

VRCC-3D+ has 8 RCC-8 relations and 15 occlusion relations. If R is an RCC-8 relation in 3D, and xObs_y is an occlusion relation, the VRCC-3D+ relations take the form R_xObs_y. The suffix y is not used if A is in front of B. This is apparent in Table 6-7. Not all obscuration relations are possible for each RCC-8 relation. As shown in Table 9, there

TABLE 8 OCCLUSION RELATION REPRESENTATION DISTANCE TABLE SHARED ENTRIES CORRESPOND TO SMOOTH EDGES

The distance table for VRCC-3D+ is 46x46. This table takes 10 pages to display, and hence is not shown here. The conceptual neighborhood graph was computed by using Algorithm 3 (VRCC-3D+), which is discussed in the next section. It has 46 nodes and 177 smooth transition edges.

A. Combining RCC-8 3D and Obscuration into VRCC-3D+

VRCC-3D+ has 8 RCC-8 relations and 15 occlusion relations. If R is an RCC-8 relation in 3D, and xObs_y is an occlusion relation, the VRCC-3D+ relations take the form R_xObs_y. The suffix y is not used if A is in front of B. This is apparent in Table 6-7. Not all obscuration relations are possible for each RCC-8 relation. As shown in Table 9, there

TABLE 8 OCCLUSION RELATION REPRESENTATION DISTANCE TABLE SHARED ENTRIES CORRESPOND TO SMOOTH EDGES

The distance table for VRCC-3D+ is 46x46. This table takes 10 pages to display, and hence is not shown here. The conceptual neighborhood graph was computed by using Algorithm 3 (VRCC-3D+), which is discussed in the next section. It has 46 nodes and 177 smooth transition edges.

A. Combining RCC-8 3D and Obscuration into VRCC-3D+

VRCC-3D+ has 8 RCC-8 relations and 15 occlusion relations. If R is an RCC-8 relation in 3D, and xObs_y is an occlusion relation, the VRCC-3D+ relations take the form R_xObs_y. The suffix y is not used if A is in front of B. This is apparent in Table 6-7. Not all obscuration relations are possible for each RCC-8 relation. As shown in Table 9, there

TABLE 8 OCCLUSION RELATION REPRESENTATION DISTANCE TABLE SHARED ENTRIES CORRESPOND TO SMOOTH EDGES

The distance table for VRCC-3D+ is 46x46. This table takes 10 pages to display, and hence is not shown here. The conceptual neighborhood graph was computed by using Algorithm 3 (VRCC-3D+), which is discussed in the next section. It has 46 nodes and 177 smooth transition edges.

A. Combining RCC-8 3D and Obscuration into VRCC-3D+

VRCC-3D+ has 8 RCC-8 relations and 15 occlusion relations. If R is an RCC-8 relation in 3D, and xObs_y is an occlusion relation, the VRCC-3D+ relations take the form R_xObs_y. The suffix y is not used if A is in front of B. This is apparent in Table 6-7. Not all obscuration relations are possible for each RCC-8 relation. As shown in Table 9, there

TABLE 8 OCCLUSION RELATION REPRESENTATION DISTANCE TABLE SHARED ENTRIES CORRESPOND TO SMOOTH EDGES

The distance table for VRCC-3D+ is 46x46. This table takes 10 pages to display, and hence is not shown here. The conceptual neighborhood graph was computed by using Algorithm 3 (VRCC-3D+), which is discussed in the next section. It has 46 nodes and 177 smooth transition edges.
are 46 VRCC-3D+ relations: 11 DC, 11 EC, 11PO, 1 EQ, 5 TPP, 5 TPPC, 1 NTPP, and 1 NTPPc relations.

<table>
<thead>
<tr>
<th></th>
<th>DC</th>
<th>EC</th>
<th>PO</th>
<th>EQ</th>
<th>TPP</th>
<th>TPPc</th>
<th>NTPP</th>
<th>NTPPc</th>
</tr>
</thead>
<tbody>
<tr>
<td>nObs</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nObs_c</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>nObs_e</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pObs1</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pObs2</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pObs_c1</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pObs_c2</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pObs_e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eObs</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eObs_c</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eObs_e</td>
<td></td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cObs</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cObs_c</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cObs_e</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eObs_c1</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eObs_c2</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The spatio-temporal distance between a pair of VRCC-3D+ relations becomes the composite distance x+y+z where x represents the RCC-8 distance in 3D, y represents the distance for 2D in projections, and z represents the perspective linear distance from the viewer. In VRCC-3D+, a transition is smooth if: (1) it is a smooth transition in the 3D RCC-8 spatial relationship, and (2) it is a smooth transition in the Obs15 relation, which is composed of an obscuring type transition relation in a 2D view plane and a nearer object determining transition along a linear (1D) view normal distance.

Although using these distances to create the graph does not result in a smooth transition graph, it does result in a visual snapshot model graph. It then is useful when additional heuristic information is used to determine smooth transitions. The gradual spatio-temporal change in objects leads to smooth transitions, even though the numerical distance does not necessarily yield accurate information. All the rules can be handcrafted and incorporated into the algorithm as rules. After determining the smooth transitions, edges can be labeled with uniform length and then can be ordered for interpolation.

For example, consider the source relation PO_pObs_c1 and destination relation EC_cObs as shown in Fig 7. As an application of Algorithms 3 and 4, the smooth transition path becomes <PO_pObs_c1, TPP_cOb_e, NTPP_cObs, PO_cObs_EC_cObs>.

![Figure 7. VRCC-3D relations PO_pObs_c to EC_cObs](image)

IV. ALGORITHMS FOR DETERMINING SMOOTH TRANSITION BETWEEN RELATIONS: RCC-8, OCCLUSION IN 3D, VRCC-3D+

We formally describe these ideas in the form of algorithms which are used in the construction of smooth transition graphs for visualization. Minimum length edges are not necessarily smooth transition edges. Using the smooth conceptual neighbors, there are 11 smooth transition edges between 28 RCC-8 relation pairs. Among 120 occlusion pairs, there are 26 smooth transition pairs. VRCC-3D+ has three types of smooth transitions: R_O1 to R_O2 (31), R_O to R_O (50), and R_O1 to R_O2 (96), for a total of 177 smooth transition edges out of 1035 connecting relation pairs.

Consider the RCC-8 relations as characterized with the 4-Intersection model. Let R1 and R2 be two relations. Then R1-R2 is the difference vector [IntInt, IntBnd, BndInt, BndBnd], and the Manhattan length of this difference vector is denoted by D, where D is abs(IntInt) + abs(IntBnd) + abs(BndInt)+ abs(BndBnd). The value of D ranges from 0 to 4. The value D=0 refers to itself; it means no significant change occurred from R1 to R2 (R1 ≡ R2) as a result of the movement of objects. The value D = 4 refers to non-smooth transitions such as DC to PO. As a formal technique for generating smooth transition graphs, have the following algorithms.

Algorithm 1 (RCC) (Smooth Transition Algorithm) determines whether R1 and R2 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 for CNG; otherwise, it returns false.

**Algorithm 1 (RCC)**

Given: A pair R1, R2 of relations in **RCC-8**.  
Returned: A logical value depending on whether R1 is a smooth transition of R2  
Boolean **STA_RCC(R1, R2)** // Smooth Transition Algorithm
[IntInt, IntBnd, BndInt, BndBnd] = R_1-R_2
D = abs(IntInt)+abs(IntBnd)+abs(BndInt)+abs(BndBnd)
If (D==1)
    If (IntInt==0)
        Return true
    Else
        Return false
ElseIf (D==2)
    If (IntInt==0 && IntBnd==0)
        Return true
    Else
        Return false
ElseIf (D==3)
    If (IntInt==0 && IntBnd==0 && BndInt==0 && BndBnd==0)
        Return true
    Else
        Return false
Else
    Return false
EndAlgorithm /* Smooth Transition Algorithm*/

In Algorithm 3, we can determine whether a pair of composite relations represents a smooth transition based upon what is known about smooth transitions for RCC-8 and occlusion relations. More specifically, for two relations R_1, R_2 from RCC-8 and O_1, O_2 from Obs15, the composite relation can be tested for the three possible cases: 1. R_1!=R_2, O_1!=O_2; 2. R_1=R_2, O_1!=O_2; and 3. R_1=R_2, O_1 = O_2

Algorithm 3 (RCC_Obs Integration)
Given: A pair R_1, R_2, and O_2 of relations in VRCC-3D+.
Returned: A logical value depending on whether R_1, O_2 is a smooth transition of R_2, O_2
Boolean STA_VRCC-3D+(R_1, R_2, O_1, O_2)
/* Smooth Transition Algorithm */
Let CNG_RCC8 be Conceptual Neighborhood Graph for spatial relations RCC-8
Let CNG_Obs15 be Conceptual Neighborhood Graph for Occlusion relations in Obs15
If (R_1, R_2) in CNG_RCC8
    If (O_1, O_2) in CNG_Obs15
        Return true /* R_1!=R_2, O_1!=O_2 */
    Else
        If (O_1==O_2)
            Return true /* R_1!=R_2, O_1==O_2 */
        Else
            Return false
    Else
        If (R_1==R_2)
            If (O_1, O_2) in CNG_Obs15
                Return true /* R_1=R_2, O_1!=O_2 */
            Else
                Return false
            Else
                Return false
EndAlgorithm

In Algorithm 4, we can use the CNG for smooth transitions of relations in RCC-8, occlusion relations, or VRCC-3D+ relations by specifying the appropriate set of relations, S.

Algorithm 4 (Transition Graph)
Given: a set S of relations
Returned: Find CNG, conceptual smooth transition neighborhood graph
/* Initialize conceptual neighborhood graph CNG to empty list of all conceptual neighbors */
CNG=null
/* Use generic STA that stands for STA_RCC, STA_Obs, or STA_VRCC-3D+ for corresponding relations to determine which pairs are conceptual neighbors. Retain all conceptual neighbors */
For {R_1, R_2} ⊆ S
    If STA(R_1, R_2)
        Add (R_1, R_2) to CNG
EndFor
Finally, Algorithm 5 can be adapted to RCC-8, occlusion, or VRCC-3D+ relations simply by adapting the current relation names.

Algorithm 5 (Transition Path)
Given: Relations \( R_1, R_2 \) in RCC-8, or Occlusion, or VRCC-3D+. Returned: Find a path \( P \) of conceptual neighbors from \( R_1 \) to \( R_2 \).

/* Use Smooth transition algorithm STA_RCC, or STA_Obs, or STA_VRCC-3D+ 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

\( S = \{ R_1 \}, T = T - \{ R_1 \} \)

Repeat

/* Find s in \( S \) and R in \( T \) that are neighbors */

If \( (s, R) \) in CNG

\( S = S \cup \{ R \}, CNG = CNG - \{ (s, R) \} \)

Elseif \( (R, s) \) in CNG

\( S = S \cup \{ R \}, CNG = CNG - \{ (R, s) \} \)

Until (CNG==null | R==R_2)

Traverse back from \( R_2 \) to \( R_1 \) through the parent edges in \( S \)

Save the path in \( P \)

EndAlgorithm

V. CONCLUSION

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 (particularly in 3D) have not received the same attention. Conceptual neighborhood paths are one way to identify spatially smooth temporal transitions of relations; they can be used to order the relations for predicting intermediate and future relations. Herein we presented a hybrid algorithm for computing smooth transitions in RCC-8, occlusion relations, and the integration of 3D spatial and occlusion relations for VRCC-3D+. This technique is simple, adaptable, and efficient. It is hoped that these algorithms will be applicable to other related studies in qualitative spatial reasoning such as GIS Scene Similarity and automated prediction of the location of spatial mobile users.

VI. REFERENCES


