

# Spatial Representation and Reasoning using the N-Dimensional Projective Approach

Jorge Pais and Carlos Pinto-Ferreira

Instituto Superior de Engenharia de Lisboa  
Instituto de Sistemas e Robótica, Instituto Superior Técnico  
Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal  
jpais@isiel.pt, cpf@isr.ist.utl.pt

## Abstract

In this paper, we introduce a novel pictorial approach for solving problems in n-dimensional Euclidean spaces called the *n-dimensional projective approach*. The projective approach is based on a hierarchical and modular architecture, where its ground module is rooted on geometrical concepts. The result is an effective and consistent spatial approach able to solve problems in n-dimensional spaces with a quadratic computational cost in order to the number of domain entities and a linear cost in order to the space dimensionality. The projective approach has been used in simulation of physical real-world problems, where physical properties of entities exercise influence on results.

## Introduction

The area of *spatial reasoning* is fruitful on research work concerning one (Detcher et al. 1991) and two dimensional spaces (Retz-Schmidt 1988). However, in what respects to higher dimensional spaces (Coenen et al. 1988) few research work has been developed. In fact, severe computational problems emerge when one or two-dimensional approaches are scaled up to higher dimensional spaces because computation time and resources grow exponentially with the number of topological relations needed to representing a domain. In qualitative spatial reasoning area some interesting research work has been developed (Cui et al. 1992) (Gots 1996), but the effectiveness of this approach decreases exponentially in order to the number of domain entities (Nebel 1995). The authors had developed reasoning techniques to improve the effectiveness of the qualitative spatial reasoning process (Pais and Pinto-Ferreira 1998). The usual approach to reasoning about space starts to define a symbolic language  $L$  (Cui et al. 1992) that includes spatial relations between pairs of domain entities based on *connect* relationship (Clarke 1981). Based on these symbolic concepts, a spatial reasoning process should be developed, which must respect fundamental spatial properties like *continuity* within a predefined *granularity*. The projective approach is a hierarchical and modular architecture, where its projective representation is defined in low levels and the reasoning process

is implemented by higher levels of the projective architecture(see Figure 1).

The remaining of this paper is structured as follows. In section 2, we make a complete level description. Section 3 provides examples about a *puzzle* domain. Finally, section 4 describes conclusions and future work.

## The N-Dimensional Projective Architecture

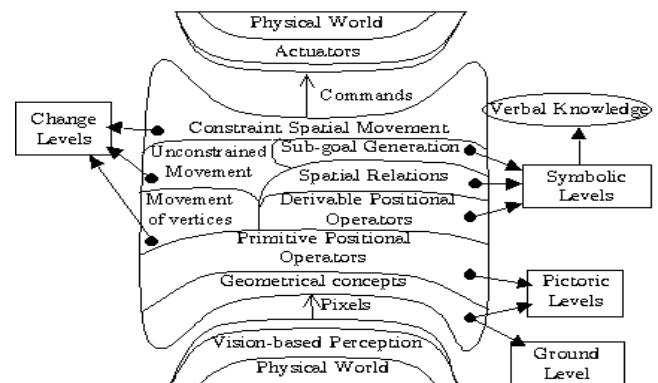


Figure 1: The *projective* architecture.

## Geometrical Concepts –Ground Projective Level

The *ground* level defines the projective representation foundations that are based on both kinds of concepts the topological like *region* and the geometrical like *projective axis*, *projective region vertex* and *projective axis vertex* (Pais and Pinto-Ferreira 2000). A *region* results from a topological transformation of the original shape of a body into a bounding box with edges parallel of all projective axis. Each domain or world representation includes as many regions as spatial entities exist in real-world  $R = \{r_1, r_2, \dots, r_k\}$ . A n-dimensional space  $S_n$  is defined by an ordered set of  $n$  *projective axis*  $S_n = \{A_1, A_2, \dots, A_n\}$ . Each projective axis is an ordered set of projective axis vertices  $A_i = \{\vartheta_1^i, \vartheta_2^i, \dots, \vartheta_m^i\}$ . Given each projective axis must be a non-empty set of projective region vertices  $\vartheta_d^i = \{V_{s,1}^i, V_{e,2}^i, \dots\}$ . A region is identified in each projective axis by a line segment delimited by a start vertex (e.g. the start vertex  $V_{s,1}^i$  of  $r_1$  upon the projective axis  $A_i$ ) and an end vertex (e.g. the end vertex  $V_{e,2}^i$  of  $r_2$  on  $A_i$ ).

## Primitive Positional Operators –Second Projective Level

The *primitive positional operators*  $\{\ll, \equiv\}$  define a minimal model of the world upon a projective axis  $A_i$  and they have the following meaning:

$V_{s,1}^i \ll V_{s,2}^i$  iff  $V_{s,1}^i$  is closer than  $V_{s,2}^i$  from the projective axis origin.

$V_{e,1}^i \equiv V_{e,2}^i$  iff both end vertices of regions  $r_1$  and  $r_2$  are equidistant from the projective axis origin.

## Derivable Positional Operators –First Symbolic Level

The *derivable positional operators* define the first symbolic level of the presented architecture. These operators must be responsible to introduce spatial semantics to the projective representation and they are asserted using the primitive positional operators, as follows:

$left(V_{s,r}^i) = \varphi_{s,r}^i \Rightarrow \forall V_{x,m}^i \in \varphi_{s,r}^i : V_{x,m}^i \ll V_{s,r}^i$

$right(V_{s,r}^i) = \delta_{s,r}^i \Rightarrow \forall V_{x,m}^i \in \delta_{s,r}^i : V_{s,r}^i \ll V_{x,m}^i$

$coincident(V_{s,r}^i) = \gamma_{s,r}^i \Rightarrow \forall V_{x,m}^i \in \gamma_{s,r}^i : V_{x,m}^i \equiv V_{s,r}^i$

## Spatial Relations –Second Symbolic Level

This level includes the spatial relations set  $\{\text{OutsideLeft}, \text{OutsideRight}, \text{OutsideLeftCoincident}, \text{OutsideRightCoincident}, \text{CompletelyCoincident}, \text{CompletelyInside}, \text{InsideLeftCoincident}, \text{InsideRightCoincident}, \text{OverlappedLeft}, \text{OverlappedRight}\}$  applied to a region. These spatial relations are presented formally in (Pais and Pinto-Ferreira 2000), which are an extension set of the relations presented in (Clarke 1981) about the calculus of individuals. One important purpose of this level is to provide a decoding system from verbal knowledge (symbolic definitions) into projective geometrical concepts and *vice-versa*.

## Sub-Goal Generation –Third Symbolic Level

This level is responsible for the sub-goal generation (possible future views) reachable from the current state (present view world) respecting spatial constraints. When we talk about spatial constraints we understand them as physical properties of obstacles or entities in system environment. For example, an embodied system either could acquire (e.g. high reflections on signal sonar could signify an *impenetrable* entity, high red detection could imply *untouchable* entities) or could integrate (e.g. white colour could mean an *impenetrable* wall, red could be an *untouchable* fire) this sort of knowledge. This level provides a real-time generation of consistent sub-goals based on a method named *projective sub-goal generation* (PSG). The PSG method classifies each projective axis using three parameters, *current state*, *transition* from current state to goal state and *goal state*. Each one of these parameters could take two different values for each physical property, violation(V) or non-violation(N).

The evaluation result for each projective axis determines the set of sub-goals reachable from the current state respecting spatial constraints as is shown in Table 1. Conditions

Current State	Current $\rightarrow$ Goal	Goal State	Sub-goal State for each $A_i$
N	N	N	Goal State
N	N	V	Impossible
N	V	N	Goal or Current
N	V	V	Goal or Current
V	N	N	Goal State
V	N	V	Permutations of Current
V	V	N	Goal State
V	V	V	Permutations of Current

Table 1: The *projective sub-goal generation*(PSG) method.

(N, N, N), (V, N, N) and (V, V, N) describe topological configurations where the goal state is the next sub-goal. The condition (N, N, V) is an inconsistent condition that never happens in a consistent domain. With respect to (N, V, N) and (N, V, V) the method provides an evaluation of the topological configuration over all other projective axis. If there exists at least one projective axis that does not occur a property violation then the next sub-goal can be defined as the goal state; otherwise, it is the current state. With regards to the critical conditions (V, N, V) and (V, V, V), both conditions do not provide any clue to solve the property violation. However, this problem can be solved using a complete sub-goal generation that is based on both steps, the generation of all permutations among regions responsible for the property violation and the combination between each one of these permutations and the non-violating regions. These permutations might be interpreted as a breadth step into the reasoning process that essentially should be based on depth steps to find out solution paths with *effectiveness*. However, breadth steps can be transformed in depth steps if the system generates one permutation at a time and memorizes these points on the reasoning process as backtracking points. Two spatial properties of entities will be considered, *untouchable* and *impenetrable*. These two properties create spatial constraints on the movement of spatial entities and consequently a proposed problem has different solutions to respect the *spatial constraint satisfaction*.

```

UntouchViolationToState( $A_i, LR$ )
{ FOR each one  $r_x$  of  $LR$  DO
  { IF ( $CompletelyInside(A_i, r_x) \cup$ 
     $CompletelyCoincident(A_i, r_x) \neq \emptyset$ ) return V;
    IF ( $OutsideRight(A_i, r_x)$ ) return V;
    IF ( $OutsideLeft(A_i, r_x)$ ) return V;
    IF ( $OutsideRightCoincident(A_i, r_x)$ ) return V;
    IF ( $OutsideLeftCoincident(A_i, r_x)$ ) return V;
  } return N; }

```

Figure 2: The *untouch* property violation within a state.

The *untouchable* property violation algorithm applied to a projective axis  $A_i$  under a domain with a list of *untouchable* regions  $LR$ , is shown in Figure 2. Algorithm presented in Figure 2 handles both states current and goal addressed in PSG method.

The algorithm shown in Figure 3 detects the *untouchable* property violation on a transition between two states. Note that in Figure 3,  $CA_i$  and  $FA_i$  represent the same projective axis  $A_i$  on current and goal times.

Another physical characteristic addressed here is the *impenetrable* property, which makes a region incapable of be-

```

UntouchViolationToTransition( $CA_i, FA_i, LR$ )
{ FOR each one  $r_x$  of  $LR$  DO
  { IF ( $OutsideRight(CA_i, r_x) \neq OutsideRight(FA_i, r_x)$ )
    return V;
    IF ( $OutsideLeft(CA_i, r_x) \neq OutsideLeft(FA_i, r_x)$ )
    return V;
  } return N; }

```

Figure 3: The *untouch* property violation on a *transition*.

```

ImpenetrableViolationToState( $A_i, LR$ )
{ FOR each one  $r_x$  of  $LR$  DO
  { IF ( $CompletelyInside(A_i, r_x) \neq \emptyset$ ) return V;
    IF ( $OutsideLeftCoincident(A_i, r_x) \cap$   

 $OutsideRightCoincident(A_i, r_x) \neq \emptyset$ ) return V;
  } return N; }

```

Figure 4: The *impenetrable* violation within a state.

ing pierced through another region. Assuming this definition, the algorithms shown in Figures 4 and 5 detect this property violation in both situations, within a state and on a transition between two states applied to the same projective axis.

```

ImpenetrableViolationToTransition( $CA_i, FA_i, LR$ )
{ FOR each one  $r_x$  of  $LR$  DO
  { IF ( $OutsideLeftCoincident(CA_i, r_x) \neq$   

 $OutsideLeftCoincident(FA_i, r_x)$ ) return V;
    IF ( $OutsideRightCoincident(CA_i, r_x) \neq$   

 $OutsideRightCoincident(FA_i, r_x)$ ) return V;
  } return N; }

```

Figure 5: The *impenetrable* violation on a *transition*.

Note that, all algorithms are easily expanded to a  $n$ -dimensional space for applying each one of these algorithms repeatedly for all  $A_i$  in domain's model.

A general topological property that should be stressed is that, physical properties like the precedent one's just are violated in a  $n$ -dimensional projective space when for all projective axis happen the property violation. Assuming that, the hierarchical projective architecture behaves effectively and consistently if the initial topological world description guarantees a non-violation of properties.

For example, considering the two dimensional problem illustrated in Figure 6. And also considering that none of those regions share any physical property. Then this problem does not have spatial constraints and consequently the higher level of our architecture could be applied without a functional intervention of this level. It means that the final state is the only sub-goal's problem.

However if we introduce spatial constraints in the model, for example considering regions  $r_1$  and  $r_2$  untouchable then this level is able to produce all sub-goals shown in Figure 7. Each sub-goal is given to higher levels when they ask by another sub-goal in both cases of failure or backtracking. At the middle-left of Figure 7 we start to draw the initial state and the evaluation result of the impenetrable property violation in order to (Current state, Transition, Goal State) for each projective axis. This evaluation is shown closer to each projective axis. After that, the PSG method gives as sub-goal results all topological descriptions presented at right of the initial state. The sequence of sub-goals that define a sub-goal path from the initial state to the goal state are called

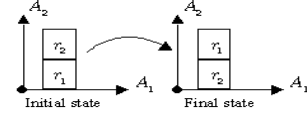


Figure 6: A two-dimensional *puzzle* problem.

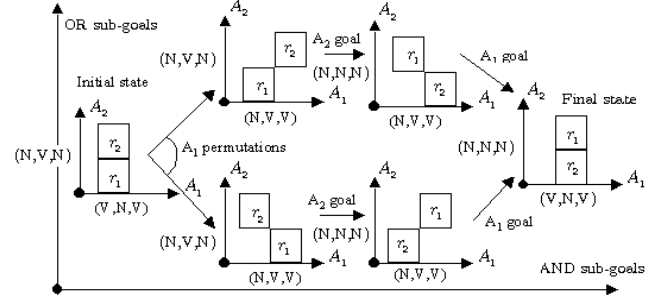


Figure 7: AND-OR sub-goals.

AND sub-goals. As you can see in Figure 7, each sub-goal path guarantees spatial constraint satisfaction. OR sub-goals define start points to alternative sub-goal path solutions for the reasoning process.

A careful analysis of the truth table presented in Table 1 and the results shown in Figure 7 ensures empirically that the PSG methodology generates sub-goals with two important characteristics, they should be as close as possible to the goal state and the transitions among them never violate domain properties.

### Movement of Vertices –First Change Level

Just two atomic movement operators are able to generate change over each projective region vertex along each projective axis.

$MoveVertexLeft(V_{x,r}^i, \vartheta_n^i)$  moves a projective region vertex  $V_{x,r}^i$  from the current projective axis vertex  $\vartheta_n^i$  to its left projective axis vertex.

$MoveVertexRight(V_{x,r}^i, \vartheta_n^i)$  changes a projective region vertex  $V_{x,r}^i$  from the projective axis vertex  $\vartheta_n^i$  to the closer projective axis vertex states on its right.

These operators are blind in order to respecting spatial constraints, because they are just based on pictorial levels of knowledge (Pais and Pinto-Ferreira 2000).

### Unconstrained Movement –Second Change Level

The one-dimensional unconstrained movement algorithm takes advantage of both level functionality, the spatial relations and the movement vertices defined on precedent sections.

Considering  $C_1$  and  $F_1$  as being the current and final projections over the unique projective axis  $A_1$  this algorithm is shown in Figure 8. A problem with  $K$  regions then each projective axis includes  $2 \times K$  projective region vertex. Then in worst case, the number of steps of this algorithm that needs to move a projective region vertex from one extreme to another extreme over the projective axis are  $2 \times K$ . As result, the worst case complexity values  $O(2 \times K \times 2 \times K)$  that is a quadratic value in order to the number of projective region vertex. The  $n$ -dimensional algorithm  $NDimProjectiveMove$  just requires executing the one-dimensional algorithm

```

OneDimensionalSpatialMove( $C_1, F_1$ )
{WHILE ( $C_1 \neq F_1$ )
  FOR  $\forall v_j^1 \in C_1$ 
  FOR  $\forall V_{x,r}^1 \in v_j^1$ 
  IF ( $V_{x,r}^1 \in F_1$ )
  {IF ( $Left(V_{x,r}^1 \in C_1) \subset (Left(V_{x,r}^1 \in F_1))$ )
    MoveVertexRight( $V_{x,r}^1, v_j^1$ );
  ELSE
    IF ( $Right(V_{x,r}^1 \in C_1) \subset (Right(V_{x,r}^1 \in F_1))$ )
      MoveVertexLeft( $V_{x,r}^1, v_j^1$ );
    ELSE
      { $(x=s)? y=e: y=s;$ 
        IF ( $Left(V_{y,r}^1 \in C_1) \subset (Left(V_{y,r}^1 \in F_1))$ )
          MoveVertexRight( $V_{x,r}^1, v_j^1$ );
        ELSE
          IF ( $Right(V_{y,r}^1 \in C_1) \subset (Right(V_{y,r}^1 \in F_1))$ )
            MoveVertexLeft( $V_{x,r}^1, v_j^1$ ); } } }

```

Figure 8: The unconstrained movement algorithm.

so many times as the number of projective axis existing on domain model. Consequently the resulting complexity of the n-dimensional algorithm is  $O(N \times 2^2 \times K^2)$ . However the complexity of this approach increases linearly with the spatial dimension and it is quadratic in order to the number of entities.

### Constraint Spatial Movement –Third Change Level

The action of this level is based on both functions of sub-goal generation level and unconstrained spatial movement level. The key idea is to give to the sub-goal generation level a postponing pair of states to get a complete sub-goal plan. This functionality is given by the *GetPlan* function that returns either a complete sub-goal plan between two states or an empty plan in case of failure. After that, this level provides consecutive pairs of sub-goals to the unconstrained movement level, which carry out simple vertex motions, thus making the spatial planning problem easier.

```

ConstraintSpatialMove(InitialState, FinalState)
{ Plan = GetPlan(InitialState, FinalState);
  WHILE (Plan  $\neq$   $\emptyset$ )
    NDimSpatialMove(RemoveFirstSubGoal(Plan),
      GetFirstSubGoal(Plan)); }

```

Figure 9: The algorithm of constraint spatial move.

The algorithm that implements this sequence of ideas and underlies this projective level is shown in Figure 9. Note that, *RemoveFirstSubGoal* function performs two actions, at first it updates the plan performing the elimination of its first sub-goal and at second it returns this sub-goal. But, *GetFirstSubGoal* function just returns the first sub-goal of a plan. For example, consider the upper sub-goal plan shown in Figure 7. The complete spatial motion plan designed by

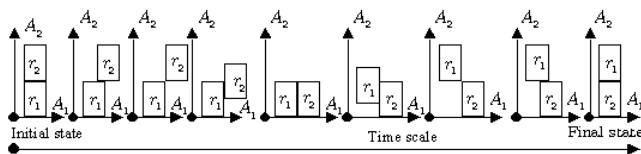


Figure 10: A plan solution for the puzzle problem. this level is illustrated in Figure 10. This Figure demon-

strates a complete simulation solution for a practical problem with just two regions, for the sake of article length. However, we simulate various problems with twenty or even a hundred of regions and the system effectiveness does not seem sensible to this increase. Thus, in practice and from an empirical point of view we had confirmed the theoretical complexity of the PSG method and in general the architecture effectiveness.

### Conclusions and Future Work

This paper provides a novel spatial approach just based on pictorial concepts and also introduces a new view of space granularity – *the projective axis vertex*. From the projective architecture emerges the potential to solve problems into a *constraint Euclidean space* without a generation of inconsistent topological descriptions that implies *effectiveness* and *computational adequacy*. Examples in *puzzle* problems domain illustrate the first promising results of this approach to solving real problems with effectiveness and continuity on spatial change. In the future, other physical properties should be developed and incorporated in order to reach a more rich representation, such as, the gravity, the dimension, the shape, etc. When all of this research work will be concluded, it could be applied in broad areas that can go through robotics, spatial reasoning, assembly planning, scheduling and vision.

### References

- Clarke, J.F.1981. A Calculus of Individuals Based on Connection. Journal of Formal Logic 2, 204-218.
- Detcher, R.; Meiri, I.; Pearl, J. 1991. Temporal Constraint Networks. Artificial Intelligence 49, 61-95.
- Retz-Schmidt, G. 1998. Various Views on Spatial Proposition. A.I. Magazine, 9(2), 95-105.
- Coenen, F.; Beattie, B.; Shave, M.; Bench-Capon, T.; Diaz, B. 1988. Spatial Reasoning Using the Quad Tesseral Representation. Artificial Intelligence Review 12, 321-243, Kluwer Publishers, Netherlands.
- Ayres, F. 1967, Theory and Problems of Projective Geometry. Schaum's outline series in Mathematics, McGraw-Hill.
- Cui, Z.; Cohn, A.; Randell, D. 1992. Qualitative Simulation Based on a Logical Formalism of Space and Time. In Proceedings of AAAI92, AAAI Press.
- Gots, N.M. 1996. Topology From A Single Primitive Relation: Defining Topological Properties and Relations In Terms of Connection. Research Report Series, Report 96.23, University of Leeds, England.
- Nebel, B. 1995, "Computational properties of qualitative spatial reasoning: First results. In Proceedings of the 19th German AI Conference, German.
- Pais, J.; Pinto-Ferreira, C. 1998. Search Strategies for Reasoning about Spatial Ontologies. In Proceedings of 10th IEEE International Conference On Tools with Artificial Intelligence, (ICTAI98), Taiwan.
- Pais, J.; Pinto-Ferreira, C. 2000. Qualitative Spatial Reasoning using a N-Dimensional+ Projective Representation. In Proceedings of 15th European Meeting on Cybernetics and Systems Research, Austria.