

Qualitative Spatial Reasoning Using Algebraic Topology

Boi Faltings

Laboratoire d'Intelligence Artificielle (DI)
Swiss Federal Institute of Technology (EPFL)
IN-Ecublens, 1015 Lausanne, Switzerland
Tel. +41-21-693-2738, FAX: +41-21-693-5225
E-Mail: faltings@lia.di.epfl.ch

Abstract. Many propositions in spatial and qualitative reasoning can be modelled as regions in phase or configuration spaces. Deciding the consistency of k propositions then amounts to deciding the question of whether there exists a point which simultaneously falls into all k corresponding regions. A more difficult problem is to decide whether there is a point which falls *only* within the given k regions. I call this *feasibility* of the set of propositions.

In this paper, I present a method for deciding consistency and feasibility for convex regions using only topological inference. It uses Helly's theorem to decide consistency of any set of k propositions based on information about consistency of small subsets. Using methods of algebraic topology, I show a sufficient method to compute a minimal skeleton of *feasible* places which accurately models the connectivity between feasible environments.

The method has been implemented. I show how to formulate and solve the piano-movers problem, an important problem in spatial reasoning, using the framework.

1 Introduction

Consider the two example problems shown in Figure 1. Example a) might arise in a mystery story: during construction work, a body was found at a spot x which is known to have been visible from the north side of the house which used to stand there. The previous owner vaguely remembers once having seen the gardener dig a suspicious hole on the east of the house. The question is: might the gardener be the killer, i.e. could the hole he dug be the place where the body was found? In Example b), the problem might be that we know certain operating regions of a chemical plant, and have to shut it down safely from a current operating point x without provoking either a boiler overheat or a fire.

Both problems can be formulated qualitatively using a set of propositions involving the position of the hole (a) or the operating point of the plant (b). More formally, let the state of a system be expressed by a continuous variable x and its model be a set of qualitative propositions $\{p_i(x)\}$. Each proposition is then associated with a region $\{r(p_i)\}$ such that $p_i(x)$ is true if and only if the state $x \in$

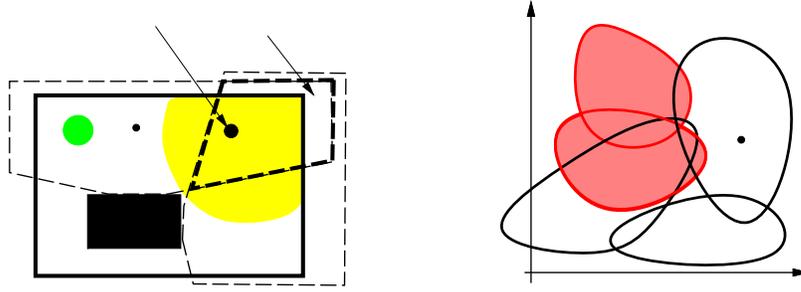


Fig. 1. Examples of problems expressible by convex regions in two-dimensional space: a) position of a point, b) operating regions.

$r(p_i)$. Thus, in example a), the proposition **visible-to-north**(x) corresponds to a region outlined by dashed lines, and in example b), **boiler-overheat**(x) corresponds to another region. In most previous work, regions are intervals of real numbers. In this paper, I consider general regions in d -dimensional space, but with the restriction that the individual regions are *convex*.

I assume that the precise geometry of regions is unknown, i.e. the pictures shown in Figure 1 are not available as such. Instead, through natural language descriptions or measurements, *overlap relations* among small subsets of regions are given. In d -dimensional space, I require knowledge of all simultaneous overlaps of $d+1$ regions and call these $(d+1)$ -relations; for example, in 2-dimensions, all simultaneous overlaps of 3 regions are given. In the example of the pond, we might know that there is a pole in the middle of the pond which is visible both to the north and to the east of the house. This means that there must be a simultaneous intersection between the 3 regions: **visible-to-north**, **visible-to-east** and **pond**. Further relations might be given by the fact that the pond touches both the northeast corner of the house and the east and north courtyard walls, and other information about the relative locations of points. In the example of the chemical plant (b), experience might indicate that some situations in the shutdown regime also make the boiler overheat and the load catch fire, so that there is an overlap between these operating regions.

I define an *environment* E as a conjunction of a set $\{p_i\}$ of unnegated propositions. Associated with the environment is a place P such that the environment is true whenever the state falls within P . P is the intersection of all regions corresponding to propositions in E : $P(E) = \bigcap_{p_i \in E} r(p_i)$. In example a), the environment $E = \{\text{visible-to-north}(x), \text{visible-to-east}(x)\}$ is true of all positions x which fall within the intersection of the regions **visible-to-north** and **visible-to-east** (shown by the bold dashed lines in Figure 1).

An environment and its associated place is called *consistent* if there exists a

point satisfying all propositions in E , which is equivalent to the condition that the associated place is a nonempty region. In example a), E is consistent, but the environment $E' = \{\text{visible-to-east}(x), \text{tree}(x)\}$ does not correspond to any place and is therefore inconsistent. An environment is called *feasible*, a stronger condition than consistency, if there is a point satisfying *only* the propositions in it. In example a), E is not feasible, since all positions x which satisfy E also satisfy the proposition $\text{pond}(x)$. In most applications, feasibility is more important than consistency. In our mystery story, my theory tells us that it is not possible to have dug one and the same hole both to the north and to the east of the house, i.e. the gardener is maybe not the killer after all. In process control, feasibility is important to know that the plant can be safely shut down by passing first through the **condensation** mode.

Somewhat surprisingly, it is possible to decide consistency and often also feasibility without knowledge of the precise shapes of the regions. I first give a simple algorithm for deciding consistency of any environment based on $(d+1)$ -relations only. The main result of this paper is an algorithm for finding a minimal *skeleton* of *feasible* environments such that (i) every feasible environment is a superset of an environment in the skeleton, and (ii) any feasible trajectory is modelled by a sequence of adjacent feasible places. The skeleton is again obtained using only topological inference on the $(d+1)$ -relations. In process control, (b) in (Figure 1), this skeleton could be used to find a path for bringing the process to a certain state while avoiding dangerous situations. In section 4, I show how a robot motion planning problem can be formulated and solved using convex regions and their minimal skeleton.

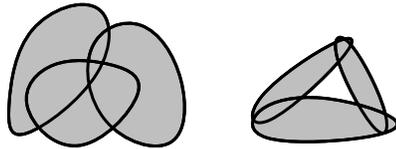


Fig. 2. *Situation A and B have the same pairwise relations, but the intersection of all three regions is consistent only in A.*

This work has been inspired in part by work on the logic of binary spatial relations ([2, 3, 4]). In these approaches, situations are modelled as conjunctions of *binary* relations among regions. However, in situations with more than 2 dimensions, such a representation is insufficient to answer queries about the consistency and feasibility of environments of propositions, as shown by the example of Figure 2. However, reasoning with binary relations can often be done in polynomial time, and may also be used for tasks such as qualitatively simulating a dynamic process [3].

Section 2 of this paper describes the theory and algorithms for deciding consistency of a set of regions. Section 3 gives an algorithm for computing homology groups which is required for the method of Section 2. Section 4 formulates the piano-movers problem as a problem of qualitative spatial reasoning, and shows the solution obtained using the methods I describe.

2 Deciding Consistency, Feasibility and Connectedness

In this section, I present three algorithms for (i) finding all maximal consistent (and feasible) environments, (ii) finding all minimal feasible environments, and (iii) finding adjacencies between environments.

Maximal Consistent Environments. An environment is consistent if the corresponding regions have a simultaneous intersection. This can be decided from the (d+1)-relations using Helly's theorem ([1]):

Theorem 1. *A set \mathcal{R} of n convex regions r_i in d -dimensional space has a common intersection if and only if all subsets of $d + 1$ regions $\in \mathcal{R}$ have an intersection.*

The set of all consistent environments can be summarized by the set of *maximal* environments, consistent environments for which there is no superset which is also consistent. Note that any maximal environment is also *feasible*, i.e. there must exist a point where only the propositions in E hold. All subsets of a maximal environment are also consistent, but not necessarily feasible.

Minimal Feasible Environments. Recall that I call an environment E *feasible* if there is a point satisfying *only* the propositions in E and no others. Contrary to consistency, feasibility of an environment E implies neither that any superset nor that any subset of E is also feasible. Furthermore, I will show below that it is impossible to compute feasibility of an arbitrary environment using only local (d+1)-relations.

Topological inference is sufficient to compute all *minimal* feasible environments using only (d+1)-relations. In many reasoning tasks where feasibility is important, the minimal environments are in fact the most interesting ones, as they represent those solutions which require a minimal set of assumptions. Furthermore, the minimal feasible environments are equivalent to the maximal consistent environments of negated propositions: if E is minimal feasible, $\bigcup_{p \notin E} \neg p$ is a maximal consistent set of negated propositions.

The algorithm for computing all minimal feasible environments works under the following assumptions:

- the union of regions in the universe U is a simply connected region without holes.
- the union of regions is bounded by a *frame* of regions which are never part of any query, but for which all (d+1)-relations are known.

When these two assumptions hold, the presence of a feasible environment can be detected by considering the *changes* the topology of \mathcal{U} undergoes as parts are removed. Figure 3 shows the principle for computing minimal environments. It

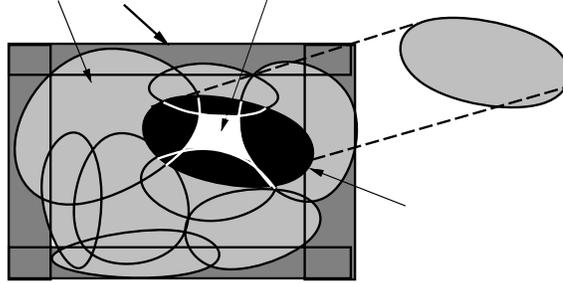


Fig. 3. The Universe \mathcal{U} is a simply connected region in configuration space, bounded by a frame. An environment E is feasible if removing its associated place $P(E)$ leaves a “hole” in \mathcal{U} . This can be detected considering the part of \mathcal{U} which overlaps $P(E)$, shown in black and called the overlap set of E .

is based on the *Alexander duality* ([7]) in algebraic topology. For the purposes of this problem, the Alexander duality states that removing a simply connected interior region P of dimension d from a simply connected space \mathcal{U} of dimension d will leave a “hole” characterized by the existence of a homology group ([7]) of rank $d - 1$. Because of the frame, all places which might be considered will always be interior, and because of convexity of the regions, they will be simply connected. For such places and environments, the following theorem holds:

Theorem 2. An environment E is feasible if and only if (1) it is consistent, (2) removal of the associated place $P(E)$ (the intersection of all regions corresponding to propositions in E) from \mathcal{U} changes the topology of \mathcal{U} from simply connected to a space with a hole, i.e. $\mathcal{U} - E$ has a non-empty homology group of rank $d - 1$.

Proof. if: Suppose that E was not feasible, i.e. any point within $P(E)$ also satisfies some other proposition p_x . Then removal of $P(E)$ cannot affect the connectivity of \mathcal{U} , since all points in $P(E)$ would be part of some other region in \mathcal{U} . Thus, the conditions imply that E is feasible.

only if: Any feasible environment E must be consistent by definition, and $P(E)$ is interior because of the existence of a frame. Suppose that removal of $P(E)$ does not leave a hole. Then, all points in $P(E)$ also satisfy some other proposition, and E is not feasible.

□

However, I do not know of any algorithm for determining the topology of the space $\mathcal{U} - P(E)$. The algorithm for deciding whether a region is simply connected, which I present in section 3, is only applicable to a union of a set of convex regions. Thus, the feasibility criterion can be applied computationally only when region $\mathcal{U} - P(E)$ can be represented as a union of convex regions. This is the case when environments are minimal, since the boundary of a minimal environment is formed exclusively by convex regions. Therefore, the method only applies to the detection of feasible environments which are also minimal.

Since the removal of the intersection of regions in $P(E)$ affects the topology of \mathcal{U} only through the parts of \mathcal{U} which actually overlap it, the way its removal affects the topology can be computed by only considering the regions of \mathcal{U} which actually overlap $P(E)$, called the *overlap set* $O(P)$ (see Figure 3). When removal of P creates a hole in \mathcal{U} , the topology of $O(P)$ must also contain this hole.

In general, an overlap set $O(P)$ whose topology is different from that of P cannot completely cover all points in P so that P must be feasible. Since each of the individual regions is convex, the topology of a place is always simply connected. Thus, all places whose overlap sets have a nonempty homology group of rank $< d - 1$ are also feasible. In fact, I show below that these environments are *join* places which make up the connections between minimal feasible environments. In Section 3, I shall give an algorithm which allows deciding whether the overlap set of a place is simply connected or not.

Connectivity. In many cases, we are considering a state space where trajectories correspond to behaviors of a device. I define a *qualitative* trajectory between two environments x and y to be a sequence of feasible environments such that there exists a precise trajectory requiring only the propositions in these environments in the given order. Furthermore, I require the trajectory to be *minimal* in the sense that each environment specifies the minimal set of propositions required to traverse the space. I define the *feasible skeleton* of the space to be the graph whose set of paths is exactly the set of minimal feasible qualitative trajectories.

When two points x and y are within the same minimal feasible place P_{min} , but possibly in different subregions, a minimal trajectory between x and y is the composition of a trajectory passing from x to a point $z \in P_{min}$ and from z to y . If place P_0 is contained in place P_{min} , any pair of points $x \in P_0$ and $z \in P_{min}$ can be connected by a path through a sequence $(P_0, P_1, P_2, \dots, P_{min})$ of places such that for all P_i , $E_{min} \subseteq E_i \subseteq E_0$, i.e. the path never traverses any regions not already contained in P_0 . Thus, there is always a trajectory from any feasible place to a minimal place such that regions are monotonically removed. For two points in the same minimal feasible environment, the only assumptions required for a trajectory between them are those which hold at the points themselves.

When the two points x and y are in different minimal environments, the trajectory between them is composed of three parts: the trajectories from x to a minimal place and to y from another minimal place, and the trajectory between minimal places. This is illustrated in Figure 4.

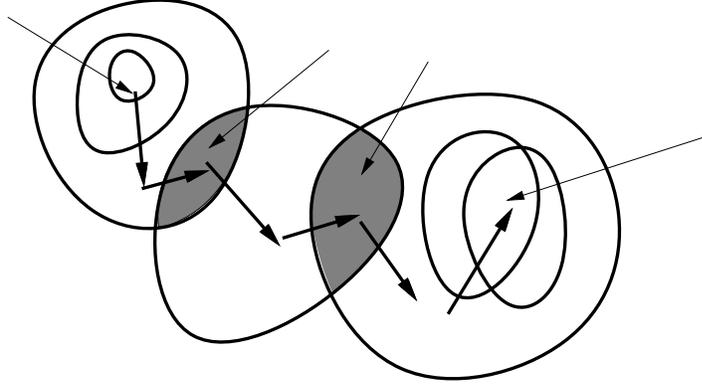


Fig. 4. *Connectivity of places. A path from x and y is composed of a path from x to the minimal place P_1 , a path along the minimal skeleton from P_1 to P_3 , and a path from P_3 to y . Note that all environments on the path from x to a point in P_1 are subsets of $E_1 \cup R_1 \cup R_2$ and require only the propositions which hold at x itself. The path from P_1 to P_3 passes through two join places (shaded), in this case $P_1 \cap P_2$ and $P_2 \cap P_3$.*

Two minimal places P_1 and P_2 are adjacent if they overlap each other, as shown in Figure 4. Moving from a point in P_1 to a point in P_2 is possible only by moving through a *join place* J , as illustrated in Figure 5, which contains at least all regions in $E_1 \cup E_2$. Note that the join place is included in both P_1 and P_2 and thus provides a minimal feasible path between the two places. Join places thus complete the minimal skeleton of the space.

Join places are identified by the following property:

Theorem 3. *The overlap set $O(J)$ of a join place connecting two adjacent minimal places P_1 and P_2 has a homology group of rank $d - 2$ or smaller.*

Proof. I refer to the situation shown in Figure 5. As P_1 and P_2 are connected via a join place J , removing P_1 , P_2 and J_i from \mathcal{U} leaves a single “hole”. Thus, the overlap set $O(P_1 \cup P_2 \cup J)$ contains exactly one homology group of rank $d - 1$. As P_1 and P_2 are only connected through the join place, removing just P_1 and P_2 from \mathcal{U} leaves two separate holes. Thus, the overlap set $O(P_1 \cup P_2)$ has exactly two homology groups of rank $d - 1$. Since $J \in (P_1 \cup P_2)$, the difference between the overlap sets $O(P_1 \cup P_2)$ and $O(P_1 \cup P_2 \cup J)$ is exactly J , and removing J from $O(P_1 \cup P_2)$ removes one homology group of rank $d - 1$. Following the theory of Mayer-Vietoris sequences (see section 3), this is possible only if the intersection contains a homology group of rank $d - 2$. But this intersection is just the overlap set $O(J)$. When $d > 2$, there may be a more complex structure where join places are themselves connected by other join places. The same reasoning then applies recursively to show that the overlap set of these places has a homology group of rank $d - 3, d - 4, \dots$ \square

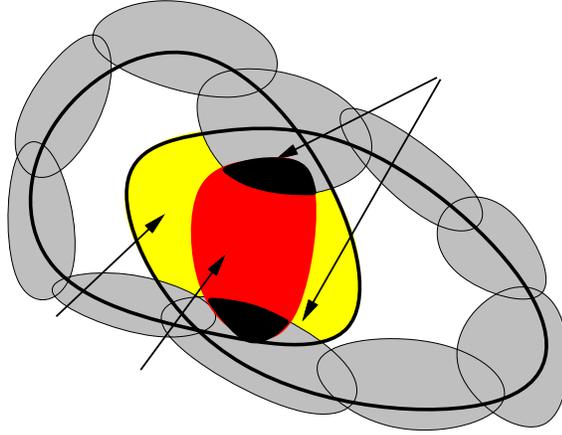


Fig. 5. Any trajectory between two adjacent minimal places must pass through a join environment. The overlap set of any join place (shown in black) must have a homology group of rank $d - 2$ or smaller. In this case, $d=2$ and so a homology group of rank $d-2$ means that the overlap set is disjoint.

Since a join place is always characterized by the presence of homology groups of rank less than $d - 1$, it can be found using a similar criterion as that used for finding minimal feasible places themselves.

Adjacencies passing through join places are now simple subset/superset relations: minimal place P is directly adjacent to join place Q if and only if $P \subset Q$ and there is no other Q' such that $P \subset Q' \subset Q$. This is the basis for computing the feasible skeleton of the universe U , a graph whose nodes are the feasible minimal, maximal and join places, and whose arcs are the adjacencies given by the subset/superset relations.

Using Minimal and Maximal Environments to Decide Feasibility. Knowledge of the minimal and maximal environments partitions the set of possible environments into three sets:

- **feasible:** minimal and maximal feasible environments found by the two algorithms above.
- **infeasible:** any environment E such that $(\exists E_{max})E_{max} \subset E$ or $(\exists E_{min})E_{min} \supset E$, i.e. which is a superset of a maximal environment or a subset of a minimal feasible environment.
- **possibly feasible:** any environment E between a minimal and a maximal environment, i.e. $(\exists E_{min}, E_{max})E_{min} \subset E \subset E_{max}$.

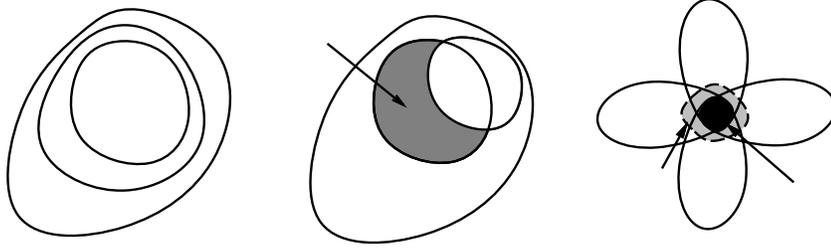


Fig. 6. The environment $(R1, R3, \neg R2)$ is inconsistent in a) but consistent in b) (shaded region). The two situations can be distinguished by characterizing the overlaps more closely. However, these relations cannot be composed. In c), no relation between 3 regions can indicate that the overlap of $(R1, R2, R3, R4)$ (black) exists only with region $R5$, i.e. that $(R1, R2, R3, R4, \neg R5)$ is inconsistent.

While consistency can be decided unambiguously, feasibility can only be decided for a subset of the environments. Figure 6 shows that the information present in the $(d+1)$ -relations is insufficient to infer more: situations a) and b) have the same $(d+1)$ -relations, but the set of feasible environments is different. This is due in part to the fact that $(d+1)$ -relations only express consistency, but not feasibility. By considering $(d+1)$ -relations based on *feasibility*, more precision can be obtained. These relations would distinguish, for example, the case where the overlap between $A \cap B$ is completely contained in C from the case where the overlap $A \cap B$ is also feasible without C . These inclusion relations are in fact a generalized version of some of the relations used in [2, 3] for *pairs* of regions.

However, as Figure 6 c) shows, these relations are not sufficient to rule out all inconsistent environments, as it can happen that the intersection of k regions falls entirely within a region R_{k+1} even though the intersection of no subset of the k regions had any containment relation with R_{k+1} . Thus, topological inference is guaranteed to find only the *minimal* feasible environments.

3 Computing Topologies

As was shown in the preceding section, the skeleton of all minimal feasible environments consists of places such that their overlap set has a topology other than simply connected. Following homology theory ([7]), the topology of a space can be represented by the dimensionalities of its homology groups. More precisely, I express the topology of a d -dimensional set of regions by a vector $(k_0, k_1, \dots, k_{d-1})$ whose i -th component is the dimensionality of the i -th homology group H_i of the space. This number is equal to the number of i -dimensional “holes” in the space: 0-dimensional holes correspond to disconnected pieces, 1-dimensional ones cut

out simply connected disks, 2-dimensional holes simply connected spheres, and so on. For a simply connected set of regions, the vector will be all zero.

I now present an algorithm for deciding whether or not the topology of a union of regions is simply connected or not. It decomposes the set by eliminating one region at a time until only a single one remains, whose topology is known to be all zero since it is a convex region. Assume that the current set of regions is $X = \{r_1, r_2, \dots, r_k\}$ and the algorithm eliminates r_j to obtain $Y = X - r_j$. Consider the overlap of r_j with the remaining regions in Y , which we call Z . Homology theory (more precisely, the theory of Mayer-Vietoris sequences [7]) shows that for every homology group of rank l eliminated from X , the overlap Z must contain a homology group of rank $l - 1$. For example, if X is a torus and Y simply connected, the removed region must have two disjoint intersections with Y . Thus, if at each elimination Z is simply connected, no homology groups are ever eliminated. Since none are present at the end, the original region was simply connected. Conversely, when X is simply connected, there is a greedy elimination order such that at every step, the intersection is simply connected.

Using the same principles, it is possible to construct an algorithm which determines the exact topology of a set of regions. However, because of possible ambiguities this algorithm might have to search different elimination orders and is unnecessarily inefficient.

To compute whether the intersection of a base place B with the union of a set of regions $R = \{r_1, r_2, \dots, r_n\}$ is simply or multiply connected, I apply the following recursive algorithm $\text{homology}(B, R)$:

1. $X \leftarrow R$
2. while $X \neq \{\}$ do
 - (a) if the intersection $B \cap \bigcap_{i=1..n} r_n$ is nonempty, i.e. a subset of a maximal consistent place, return **simply-connected**
 - (b) find $r_j \in X$ such that the overlap $r_j \cap (X - r_j)$ is simply connected, i.e. has no homology groups of any rank. The topology of the intersection is computed by calling the algorithm recursively: $\text{homology}(B', R')$ with $B' = B \cap r_j$ and $R' = O(B') \cap X$.
 - (c) if such an r_j is not found: return **multiply-connected**
 - (d) set $X \leftarrow X - r_j$
3. if B is consistent, return **simply-connected**, otherwise return **multiply-connected**

If all maximal consistent places are known, the consistency tests required by the algorithm amount to simple subset relations, and the algorithm is then very efficient, in spite of the fact that its worst-case complexity is clearly exponential.

Computing the Minimal Feasible Skeleton.

1. compute the set of maximal consistent environments \mathcal{M} .
2. compute the set of candidates for minimal feasible and join environments as all subsets shared by at least 2 environments in \mathcal{M} .

3. $m \leftarrow$ those candidates E such that $O(P(E))$ is not simply connected.
4. $\{Adj\} \leftarrow$ all pairs $E_1, E_2 \in (\mathcal{M} \cup m)$ such that $E_1 \subset E_2$, but there is no E' such that $E_1 \subset E' \subset E_2$.
5. return $(\{nodes, undirectededges\}) = (\mathcal{M} \cup m, \{Adj\})$.

4 Example: the Piano Movers Problem

As an example of an application of the formalism I developed, consider the *piano movers problem* [8], a problem which has drawn much attention in robotics and spatial reasoning. In the piano movers problem, the goal is to find a path for moving a single rigid *moving object* from an initial to a final position such that it does not collide with any of the fixed and rigid obstacles.

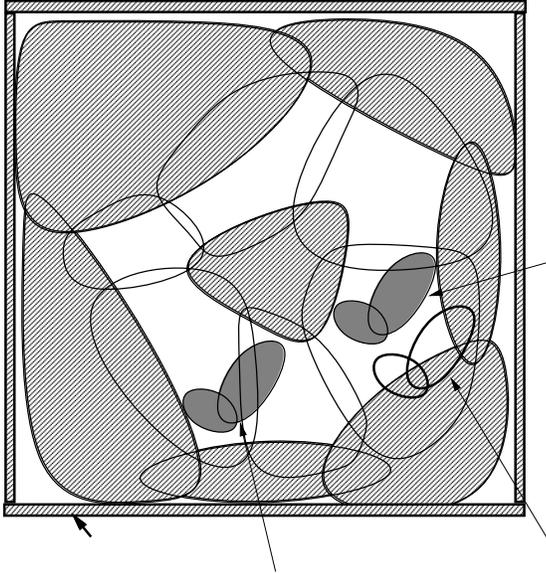


Fig. 7. Input representation of an instance of the piano movers problem. Pieces, hatched, and cavities, shown as transparent, are numbered sequentially. The moving object consists of two pieces, x and y , and is shown in three different configurations. I represent its qualitative configuration by an environment of overlaps written as a/b , where a denotes the part of the moving object and b the region it overlaps. Thus, configuration $A = \{x/8, y/8, y/9\}$, $B = \{x/10, y/10, y/11\}$ and $C = \{x/1, y/1, y/5, x/10, y/10\}$. Configurations A and B contain only bubbles and are thus legal, while C contains three obstacles and is thus not legal.

In a qualitative version of the problem, positions of the moving object are given by regions within which all points are considered equivalent. As an ex-

ample, consider the situation shown in Figure 7. It shows a *moving object*, in grey, and a set of obstacles, hatched. The moving object consists of two convex regions, labelled x and y . The free space available for moving the object is covered with convex regions called *bubbles*. I define a qualitative configuration of the moving object by the combination of regions, obstacles or bubbles which the moving object overlaps.

A *configuration* is a particular position and orientation of the moving object and defines a point in a *configuration space* ([6]), which is spanned by these parameters. Configuration space consists of blocked configurations where the moving object would overlap a fixed one, called *blocked space*, and its complement of legal positions, called *free space*. I model both spaces by a universe of two different types of convex regions: *obstacles* and *bubbles*. Each possible overlap between a part of the moving object and part of a fixed object defines a configuration space region (c-region) of illegal configurations, called an *obstacle* o_i . Blocked space \mathcal{B} is the union of all obstacles: $\mathcal{B} = \bigcup_i o_i$. Each possible overlap between a part of the moving object and a cavity defines a c-region which I call a *bubble* b_j . Note that in contrast to blocked space, free space is only a *subset* of the union of all bubbles, as all configurations falling within blocked space are excluded from it: $\mathcal{F} = \bigcup_j b_j - \mathcal{B}$

I enclose all objects by a rectangular *frame* FR , a set of regions which bounds all objects, whose purpose is to ensure consistency of the topological computation. The union of all obstacles, bubbles and frame is the universe $\mathcal{U} = \bigcup_i O_i \cup \bigcup_j B_j \cup FR$. Note that since all of physical space is covered by object parts or cavities, any configuration of the moving object falls within some c-region. The universe is therefore a simply connected region.

Because of the convexity of object parts, it is possible to show:

Theorem 4. *Every c-region formed by two convex pieces or cavities A and B is a convex region.*

Any configuration of the moving object can be represented qualitatively by the combination of region overlaps which is present in the configuration, expressed as an environment of obstacles and bubbles. Figure 7 shows examples of configurations and their representation by environments. An environment E is *legal* if P is feasible and contains only bubbles (overlaps with open space). Thus, the condition of non-overlap between moving object and obstacles can be modelled as feasibility of environments.

Feasible Skeleton. Figure 8 shows the legal part of the feasible skeleton for the example. It qualitatively represents all legal motions of the moving object. A qualitative solution to the piano movers problem can be given by first mapping the initial and final positions to the minimal places to which they belong, and then finding the qualitative path between initial and final place by searching in the graph. For example, a path between configurations A and B (Figure 7) can be found as follows. First, I map to the minimal feasible places: $A = \{x/8, y/8, y/9\}$ is already minimal, $B = \{x/10, y/10, y/11\}$ is mapped to

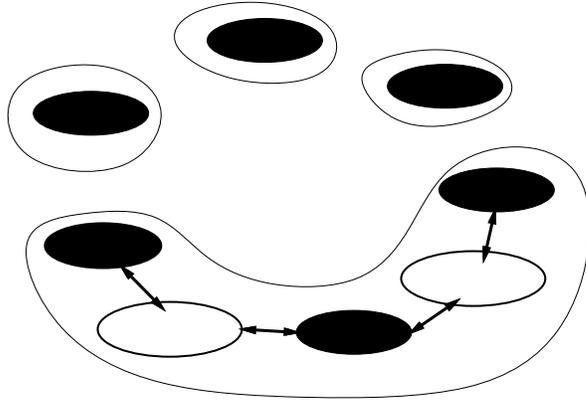


Fig. 8. *The legal part of the feasible skeleton for the example. All places containing overlaps (obstacles) have been omitted. The numbering of the regions refers to Figure 6. Minimal places are shown in black, join places in white.*

$\{x/10, y/10\}$. Next, I find a path in the graph between the two minimal environments, in this case the path $\{x/8, y/8, y/9\} \Rightarrow \{x/8, y/8, x/9, y/9, y/10\} \Rightarrow \{x/9, y/9, y/10\} \Rightarrow \{x/9, x/10, y/9, y/10\} \Rightarrow \{x/10, y/10\}$. Using more complete (d+1)-relations (as described in Section 2), for this example it would be possible to construct a complete graph containing *all* feasible environments (not only minimal ones). This graph also contains environment *B* so that no mapping to a minimal environment would be required.

I have implemented a prototype which demonstrates the topological reasoning techniques on the piano movers problem for two-dimensional objects. The input to the program is given in the form of three collections of convex bitmaps, representing the parts of the fixed objects, the moving object, and the cavities. A preprocessor uses these bitmaps to determine all possible simultaneous overlaps of 3 pairs of parts. A first version computes this by systematic search of all possibilities, which is rather slow. A much faster solution consists of detecting simultaneous overlaps in randomly generated positions of the moving object. The implementation shows that it is possible to solve such complex planning problems without *any* analytical representations.

This example shows that the minimal skeleton is powerful enough to solve problems of real-world complexity. In this example, the qualitative representation has the advantage that arbitrary object shapes can be dealt with at no penalty. Currently known techniques for solving such problems with curved shapes require approximations with algebraic curves, and extremely complex and brittle computation to determine the possible paths.

Many other applications, such as control system design ([9]) require similar forms of reasoning. It is likely that topological inference can provide similar advantages there as well.

5 Conclusions

In this paper, I have presented the concept of *feasibility* of a set of qualitative spatial propositions and given an algorithm for computing a *minimal feasible skeleton* sufficient to answer many important queries. Note that the set of feasible minimal environments is equivalent to the set of maximal environments of negated propositions: if $E_{min} = p_1 \wedge p_2 \wedge \dots \wedge p_i$ is feasible, then $N_{max} = \bigwedge_{j,j \neq 1..i} \neg p_j$ is a maximal consistent environment.

Using topological rather than geometric inference has important advantages. First, regions with curved shapes can be handled without any increase in complexity, as long as they are convex. Object representations as unions of convex parts have long been postulated in vision research ([5]), so this may not be a severe restriction. Second, the methods are simpler and more robust than geometric computation. This is because (i) they do not require computing the presence of precise points, but only the presence of regions with many points, and (ii) topological inconsistencies which may arise in geometric computation cannot occur here.

I am currently working on extending the robot motion planning application to also include rotation. Another interesting direction for further work in the context of the planning application would be to consider the use of abstractions to speed up the computation. I am also considering the applying the approach to other spatial reasoning problems.

References

1. **V. Chvátal**: "Linear Programming," W. H. Freeman, 1983
2. **D.A. Randell, A.G. Cohn, Z. Cui**: "Naive topology: Modelling the force pump," in P. Struss, B. Faltings (eds.): *Recent Advances in Qualitative Physics*, MIT Press, 1992
3. **Z. Cui, A.G. Cohn, D.A. Randell**: "Qualitative Simulation Based on a Logical Formalism of Space and Time," *Proceedings of the 10th National Conference of the AAAI*, AAAI Press, 1992
4. **M.J. Egenhofer**: "Reasoning about binary topological relations," in O. Gunther, H.J. Schek(eds.): *Advances in Spatial Databases*, pp. 143-160, Springer-Verlag, 1991
5. **D.D. Hoffman, W.A. Richards**: "Parts of Recognition," *Cognition* **18**, 1985
6. **T. Lozano-Perez, M. Wesley**: "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," *Comm. of the ACM*, **22**, 1979,
7. **E. Spanier**: "Algebraic Topology", Mc. Graw Hill, 1966
8. **J.T. Schwartz, C.K. Yap**: *Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics*, Erlbaum, Hillsdale, N.J., 1987
9. **F. Zhao**: "Phase Space Navigator: Towards Automating Control Synthesis in Phase Spaces for Nonlinear Control Systems," *Proceedings of the 3rd IFAC International Workshop on Artificial Intelligence in Real Time Control*, Pergamon Press, 1991