Topology-based Spatial Reasoning

Boi Faltings
Artificial Intelligence Laboratory (LIA)
Swiss Federal Institute of Technology (EPFL)
IN-Ecublens, 1015 Lausanne, Switzerland
FAX: +41-21-693-5225, e-mail: faltings@lia.di.epfl.ch

April 18, 1994

Abstract

Determining the possible positions and motions of objects based on their geometry is fundamental to reasoning about the physical world, for example in robot planning or mechanical design. Existing techniques are based on the geometry of object boundaries and limited in the degrees of freedom they allow, or in the object shapes that can be considered.

In this paper, I present a technique which is based on the *topology* of objects and space, and does not require a closed-form representation of object boundaries. The technique is simpler, more efficient and more robust than techniques based on geometry. However, it is limited to objects which can be represented as the union of convex subparts.

1 Introduction

Spatial reasoning about possible motion and kinematics of physical objects is fundamental for reasoning about the physical world. Figure 1 shows an example of 5 different positions of a designated *moving object* in the free space left by a set of *obstacles*. The spatial reasoning task I address in this paper is to determine a complete vocabulary of all legal qualitative positions of the moving object, called *places* ([3]), and the connectivity between these positions. Using the place vocabulary, it is for example possible to classify positions A, B, C and D as belonging to different places, and to show that positions D and E are connected, but A and C are not. For example, a place vocabulary can be used to solve the *piano-*

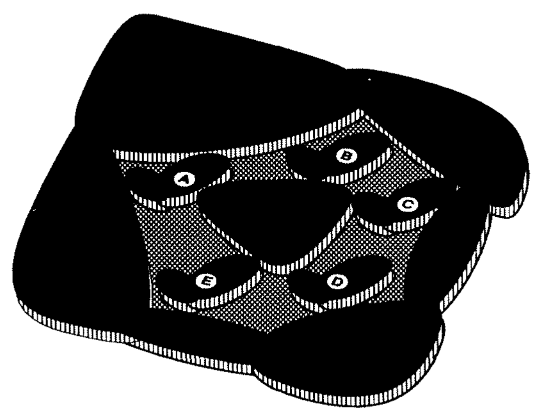

Figure 1: *Five qualitatively different positions of an object in a space of obstacles. My technique predicts the existance as well as the connectivity of the positions on the basis of the geometry of the obstacles and the moving object.*

movers problem, which has been extensively studied in the literature ([10, 6]).

Existing techniques for computing place vocabularies (among many others, [9, 3]) first model the object surfaces using equations, transfer these into *constraints* in a *configuration space* ([11]), and then obtain a region structure by algebraic methods. However, complexity imposes limitations on the allowable object shapes and their freedom of motion. Thus, no algorithm has been known which can compute place vocabularies for curved objects as shown in Figure 1. Furthermore, modeling object geometry using algebraic constraints poses several problems:

- fitting: shapes are observed as bitmaps and their automatic modeling with algebraic curves is not solved in a reliable manner.
- brittleness: small numerical errors can result in grossly incorrect topologies. Furthermore, errors propagate even to parts of the place vocabulary which were otherwise computed correctly.
- adaptability: solutions for similar situations cannot be reused.

An alternative to modelling objects algebraically by their boundaries is to model their topology as a set of regions. In this paper, I present a novel method for computing place vocabularies based on such an object model and principles of algebraic topology. Similarly to the work of Cui, Cohn and Rendell ([2]), reasoning is based on overlaps between regions. However, our work is different in that it takes into account the shapes and dimensions of rigid objects and applies algebraic topology to infer additional information. I will first define the method for the case of two-dimensional objects with two degrees of freedom, and then show how it generalizes to rotations. I have not yet investigated the generalization to three dimensions.

2 Object and constraint representation

Objects are modelled as the union of convex *parts*, and one of the objects is identified as a *moving object*. A *configuration* is a particular position and orientation of the moving object and can be defined as a point in a *configuration space* ([11]), which is spanned by these parameters. Configuration space consists of blocked configurations where the moving object would overlap others, called *blocked space*, and its complement of legal positions, called *free space*. Each possible overlap between parts of the moving object and a fixed object defines a configuration space region (c-region) of illegal configurations, called an *obstacle*. Blocked space is the union of all obstacles.

In order to be able to refer to positions of the moving object within free space, we define a set of convex regions called *cavities* which completely cover the empty space left by the fixed objects. The choice of these regions is arbitrary: they can be understood as defining a quantity space of positions of the moving object. For example, the situation of Figure 1 can be represented by the regions shown in Figure 2. The possible overlaps between a part of the moving object and a cavity defines a c-region which we call a *bubble*. Note that in contrast to blocked space, free space is only a *subset* of the union of all bubbles.

When the moving object is placed in an arbitrary configuration C, it overlaps some parts and cavities. In configuration space, this means that the configuration falls within a certain combination of corresponding obstacles and cavities, and we call such a combination an *environment*:

Definition 1 *An* environment *is a combination of c-regions, and denotes the set of configurations where the moving object exhibits at least the overlaps corresponding to them (and possibly others as well).*

The environments corresponding to two sample configurations are shown in Figure 2.

Properties of c-regions Because they represent configurations of overlap between convex regions, c-regions have the following properties which will be important for computing the topology of a combination of c-regions. Note that in this section, we consider only *translations*. Rotations will be discussed later in the paper.

Theorem 1 *Every c-region formed by two convex pieces or cavities A and B is a simply connected region.*

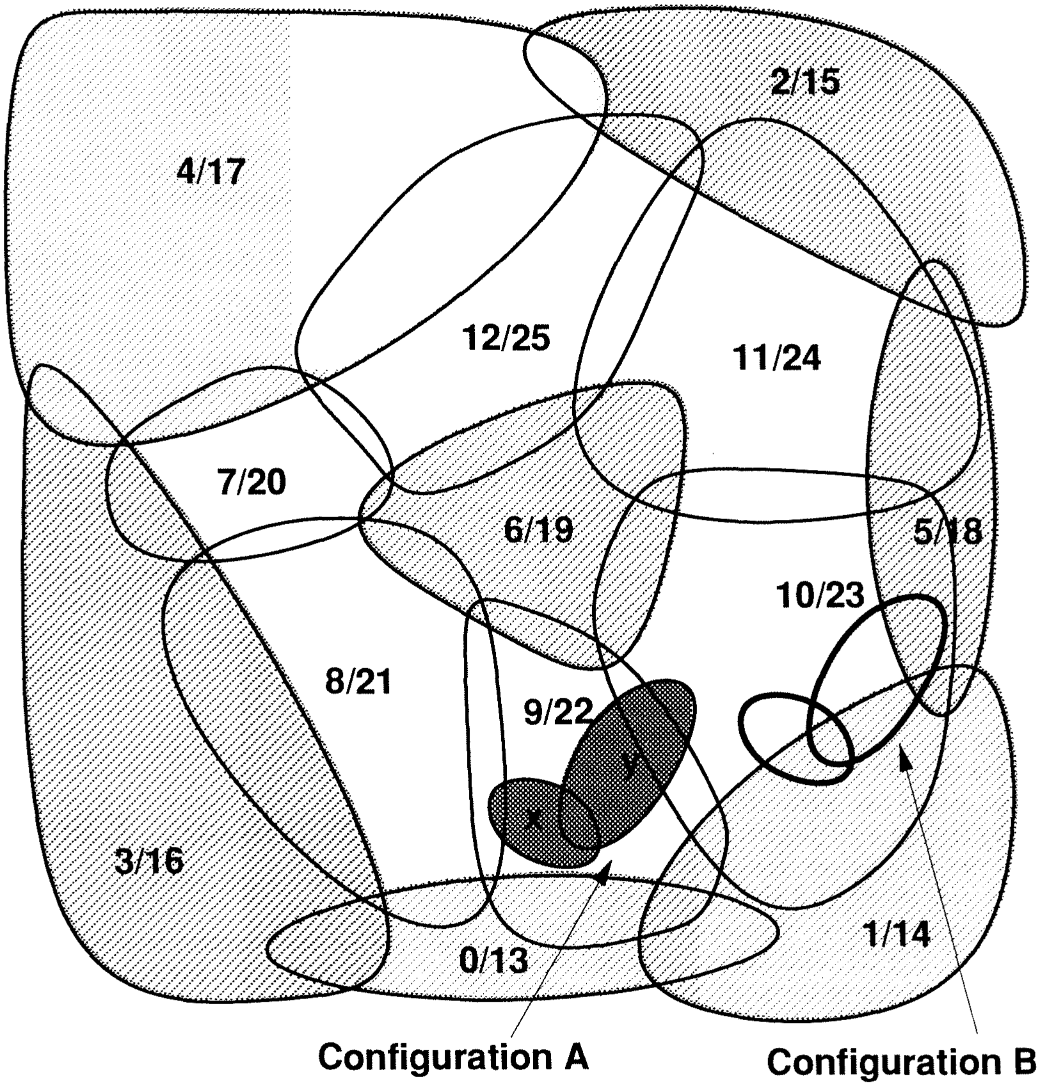

Figure 2: *Input representation of a situation. Pieces, shown in grey, and cavities, shown in white, are labelled by a combination of two numbers which number the c-regions of overlap with the two pieces x/y of the moving object, shown in grey. The shown legal configuration falls within the environment* $E_1 = \{8, 9, 22, 23\}$. *The configuration shown as an outline falls within the environment* $E_2 = \{1, 10, 14, 18, 23\}$. E_1 *contains only bubbles and is thus legal, whereas* E_2 *contains several obstacles and is illegal.*

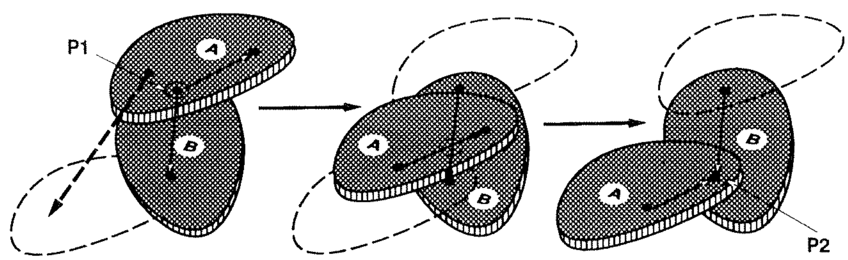

Figure 3: *We identify points $P1$ and $P2$ as* selected *overlapping points in the two shown configurations of overlap between A and B, and denote their positions on A and B by subscripts. The two configurations can be transformed into each other by moving object A in a straight line as shown. In any intermediate position, there is an overlap between a point on A which falls on $\overline{P1_A P2_A}$ and a point on B which lies on $\overline{P1_B P2_B}$. Consequently, the path where A moves in a straight line with respect to B lies completely within the c-region.*

Proof: Consider an arbitrary pair of configurations 1 and 2. Figure 3 shows a proof that there always exists a path entirely within the c-region which connects the two configurations, and thus the c-region is connected. Now consider a cycle of configurations within the c-region. The cycle can be approximated as a sequence of configurations which can be transformed into each other using the transformation shown in Figure 3. On each part, the sequence of transformations of the selected point forms a polygon whose edges all fall within the part. By shortening each translation by the same proportion ϵ, the corresponding sides of the polygon are also shortened by the same fraction and we obtain a similar polygon which is smaller by a factor of ϵ. By repeated and continuous application of this contraction operation, the polygon and thus the cycle of transformations can be contracted into a single point. Since any closed path within it can be transformed into a single point, the c-region is simply connected.

QED

Computing the topology of a set of regions requires consideration of their intersection. For this, the following property is important:

Theorem 2 *Every intersection of k c-regions $c_1, c_2, ..., c_k$ is a simply connected region.*

Proof: In both configurations A and B, let there be an overlap between the k pairs of pieces which define the c-regions. The proof of Theorem 1 shows that the straight line translation between A and B maintains each of the overlaps and thus falls within the intersection of the k c-regions. By the same reasoning as before, this intersection is thus a simply connected region.

QED

3 Environments, Cliques and Places

Recall that an environment is a combination of obstacles and bubbles. An environment E is *feasible* if there actually exists a configuration which falls *only* within E, and in particular does not intersect any other c-regions outside of E. A feasible environment thus represents a set of configurations where the moving object overlaps exactly the pieces and cavities designated by E. An environment is called *maximal* if there is no feasible environment that E is a proper subset of, and *minimal* if there is no feasible environment which is a proper subset of E. Environments which are feasible and contain only bubbles are regions of legal configurations of the moving object, and make up the *places* in the place vocabulary.

The principle underlying the method I present is the following:

> A place $P = \{B_1, B_2, ..., B_k\}$ is part of the place vocabulary only if
>
> - it is an environment consisting only of bubbles, and
> - *removing* the intersection of $\{B_1, B_2, ..., B_k\}$ leaves a "hole" in the configuration space.

which is true because if there is a configuration which falls only in P, removing P will remove the point from the configuration space, thus creating the hole. This principle allows using the topological notions of connectedness for computing place vocabularies.

An environment is the intersection of a set of c-regions and thus by Theorem 2 simply connected. A *place* represents a set of feasible positions of the moving object, i.e. positions where it does not overlap any obstacles. Thus, we define:

Definition 2 *A* place *is a feasible environment which contains only bubbles.*

The method for computing topologies is based on a theorem which allows to decide the existence of regions where any number of parts overlap from knowledge of the existence of all regions where only $d+1$ parts overlap, where d is the dimensionality of the configuration space. More specifically, we define:

Definition 3 *The* nth-order region graph $G_n(R)$ *of a set of c-regions R is the hypergraph whose nodes are the c-regions in R and whose arcs are all intersections of up to n c-regions in R.*

and

Definition 4 *An* n-clique *of a hypergraph G is a set $N = \{n_1, n_2, ..., n_k\}$ of nodes such that any subset of n nodes $\in N$ is an arc of G.*

and have the theorem:

Theorem 3 *Let G be the (n+1)-th order region graph of a set R of c-regions in an n-dimensional configuration space. The environment E consisting of an overlap of the set of c-regions N exists if and only if N is an (n+1)-clique of G.*

Proof: The "only if" direction is obvious, since an intersection of all regions in N automatically implies an intersection of all subsets of N. The "if" direction is proven inductively in following way. Obviously the theorem is true for $|N| = n+1$. Assume that it holds for some $|N| = l \geq (n+1)$, and let $N = \{n_1, n_2, ..., n_{l+1}\}$. Then all intersections of subsets of l regions exist, so all the Cěch-cohomology groups of N up to degree $l-1$ are identical to those of an intersection of $l+1$ identical unit balls in $\mathcal{R}^n$. But as a consequence of the Alexander duality, all Cěch-cohomology groups of degree $\geq n$ are identically zero. Thus, Cěch-cohomology of the N is entirely the same as that of an intersection of $l+1$ unit balls, and the intersection of all regions is non-empty.

QED

More details about Cěch-cohomology can be found in textbooks on algebraic topology, for example [8] or [7]. Theorem 3 is a generalized version of Helly's theorem ([1]), which states the same relation but for convex sets only.

For two dimensional configuration spaces, Theorem 3 implies that the set of environments formed by the c-regions is given by the 3-cliques of the region graph G. The region graph can be obtained by exhaustive testing of all possible intersections between triples of obstacles and bubbles: in the example of Figure 2, the region graph contains 26 nodes and 469 hyperarcs linking sets of three nodes.

Because the region graph is a type of intersection graph, and each 3-clique corresponds to an actual region of space, their number is limited to grow only polynomially in the number of nodes. Thus, the set of *maximal* 3-cliques of G can be determined by exhaustive search without extensive complexity; in the example we find a total of 34 maximal cliques. All these maximal cliques are feasible: configurations in them cannot be part of any other regions as otherwise the clique would not be maximal. However, most are not places since they are not composed exclusively of bubbles.

Computing feasible environments Many feasible environments are non-maximal cliques, i.e. subsets of maximal cliques, but not all subsets of maximal cliques are feasible. In order to decide whether a given non-maximal clique is feasible, we make use of the criterion mentioned in the introduction,

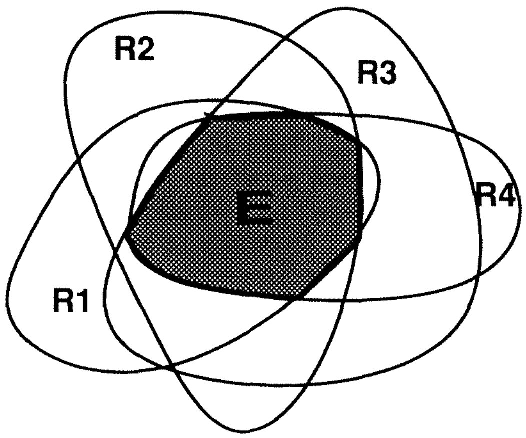

Figure 4: *An example of an environment: E is formed by the intersection of c-regions* $R1$ *through* $R4$.

namely that an environment is feasible if removing its c-regions changes the connectivity of configuration space.

More precisely, a set of c-regions E is an environment if and only if it is a subset of a maximal clique. An environment is an intersection of c-regions and by Theorem 2, its topology is always that of a simply connected region. Figure 4 shows an example of an environment formed by the intersection of c-regions $R1$ through $R4$.

We now consider the set of c-regions $O(E)$ overlapping E, called the *overlap set of* E, is the union of maximal cliques containing E, restricted to the points within E:

$$O(E) = \Big(\bigcup_{s \in S(E)} s \Big) \bigcap E$$

where $S(E) = \{s | s \text{ is a maximal clique and} s \supset E\}$.

In a two-dimensional configuration space, the overlap set of a set E can have the following topologies, illustrated by Figure 5:

a) **simply connected**: c-regions R5 and R6 cover all configurations in the environment, consequently removal will not create a hole, and the environment is **not feasible**.

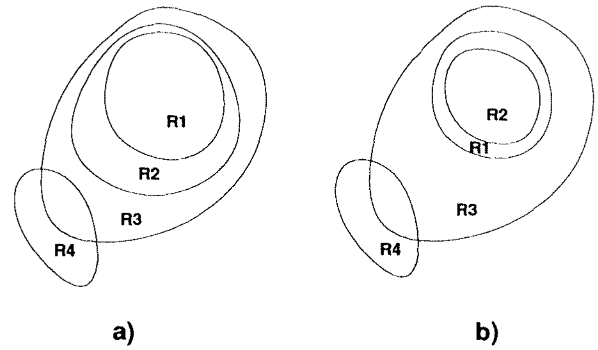

Figure 6: *Two indistinguishable situations with different environments.*

b) **simply connected**: c-region R5 covers some of the configurations in the environment, removal will create a hole, and the environment is **feasible**.

c) **multiply connected**: c-regions R5-R8 form a cycle which leaves an opening when the environment is removed, and thus the environment is **feasible**.

d) **not connected**: c-regions R5 and R6 are not connected, removal of the environment leaves an opening, and thus it is **feasible**.

The topology of the overlap can be computed using a decomposition into elementary spaces, as described later. The method for computing the feasible non-maximal environments is based on the following theorem:

Theorem 4 *If the topology of the overlap set* $O(E)$ *is different from that of the environment* E *itself,* E *is feasible.*

Proof: since $O(E)$ has a different topology from E, and $O(E) \subset E$, E must contain points which are not in $O(E)$. Thus, there are some points in E which are not in any other c-region, and thus E is feasible.

QED

This theorem leaves ambiguous the case where an environment and its overlap set have identical topologies, in this case simply connected. In fact, the formulation of region intersections does not contain enough information to distinguish whether or not such an environment is feasible. Consider the example shown

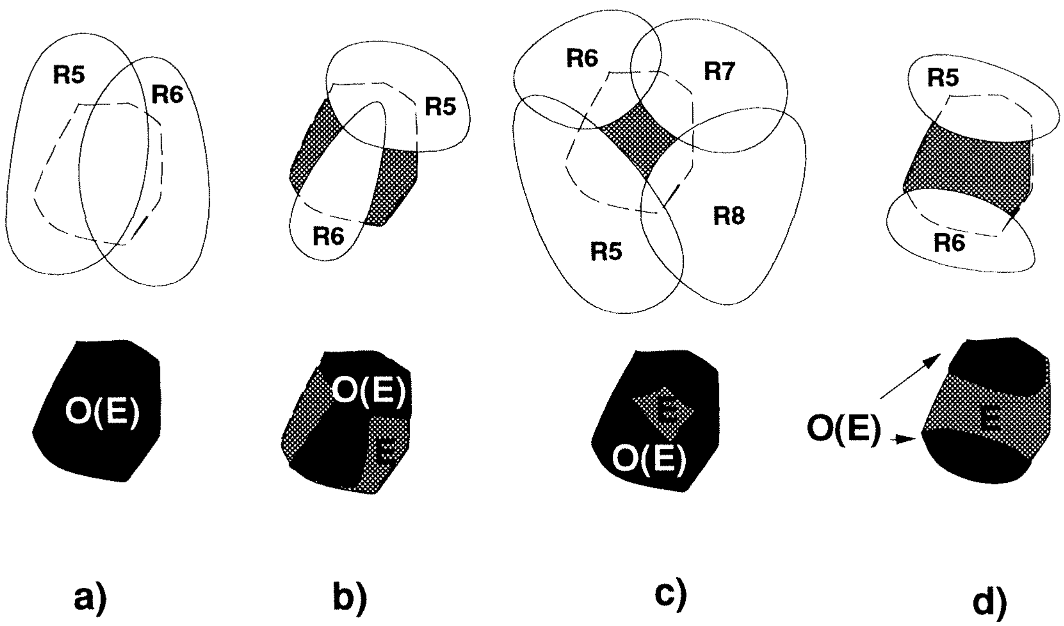

Figure 5: *Let E be the environment consisting of the intersection of the 4 c-regions R1-R4, shown in grey. Depending on the topology of the c-regions overlapping E, it may (b, c and d) or may not be (a) feasible.*

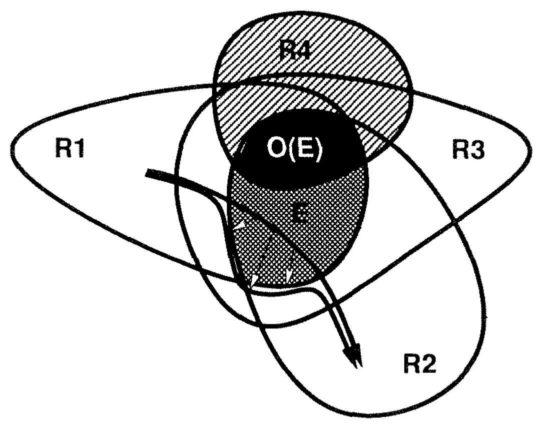

Figure 7: *E is formed by the intersection of R1, R2 and R3 and overlapped by R4. O(E) is simply connected, but does not completely cover E. Any path through E may then be transformed into a path through subset environments, in this case* $\{R1, R3\}$, $\{R3\}$, $\{R2, R3\}$. *Thus, omitting E from the place vocabulary does not cause any change in connectivity.*

in Figure 6. Cases a) and b) both have the same two maximal 3-cliques: $\{R1, R2, R3\}$ and $\{R3, R4\}$ and are thus indistinguishable in the input information given to my algorithm. However, in case a) the environment $\{R2, R3\}$ is feasible and the environment $\{R1, R3\}$ is not, whereas in case b) the opposite is true. Note, however, that all environments in question are subsets of the single environment $\{R3\}$, which is feasible by the criterion given above (overlap set not connected). Thus, both situations are *correctly modelled by the single place* $\{R3\}$, and the ambiguous environments can be ignored.

It remains to show that ignoring these environments does not result in incorrect connectivity of the space. To do this, we show that any path through an environment whose overlap set is simply connected can be transformed into an equivalent path through one of its subsets, as shown in Figure 7. We begin by showing the following lemma:

Lemma 1 *Let* E, $O(E)$ *be simply connected and* $E - O(E)$ *consist of the components* $c_1, c_2, \ldots$. *Then each* $c_i \cap Bnd(E)$, *where* $Bnd(E)$ *is the boundary of* E, *is connected.*

Proof: assume there was a component intersecting $Bnd(E)$ in two disjoint pieces. Then the boundaries of $E - O(E)$ connecting the endpoints of the pieces delimit two disjoint pieces of $O(E)$, and thus $O(E)$ is not simply connected.

QED

As a consequence, we have the following theorem:

Theorem 5 *Let* E *be an environment such that both* E *and its overlap set* $O(E)$ *are simply connected. Then any legal path through* $E - O(E)$, *the part of* E *not covered by* $O(E)$, *can be continuously transformed into a legal path through subsets of* E.

Proof: Let c be a component of $E - O(E)$. A path through c must cross the boundary of E $Bnd(E)$ an even number of times. Since both c and $c \cap Bnd(E)$ are simply connected, the crossings can all be continuously contracted into a single point and the path thus removed from c.

QED

Thus, it is correct to consider as feasible only those environments which are either maximal or whose overlap sets are not simply connected, and this is the rule my algorithm uses.

Computing the place vocabulary Those feasible environments which do not contain any obstacles, i.e. are made up purely of bubbles, are environments in which there is no overlap between moving object and fixed pieces and make up the places in the place vocabulary. In the example given earlier, there are 81 environments which are feasible by the topology of their overlap set or by being maximal, of which 9 do not contain any obstacles and thus make up the places. They are shown in Figure 8.

Transitions between places are possible at boundaries of c-regions. In particular, a transition from $P1$ to $P2$ can correspond to either entering or leaving a c-region which distinguishes $P1$ from $P2$. Thus, $P1$ is adjacent to $P2$ whenever $P1$ is a proper subset of $P2$, or $P2$ is a proper subset of $P1$. This generates the adjacency relations shown in Figure 8.

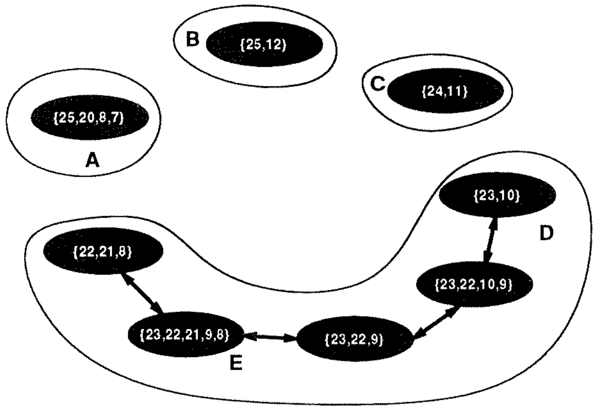

Figure 8: *Place vocabulary for the example. The numbering of the regions refers to Figure 2, the letters to configurations in Figure 1.*

4 Compositional computation of topology

For deciding whether an environment is feasible or not, it is necessary to compute the topology of its overlap set. The algorithm I use is based on constructing a decomposition of the space into subspaces of known topology. One basis for this computation is the theorem of Seifert & Van Kampen ([7]), which states that

Theorem 6 *Two simply connected regions A and B, overlapping in a simply connected region C, form a simply connected region.*

Proof:
see [7].

The other important basis is the following:

Theorem 7 *Two simply connected regions A and B, overlapping in several disjoint regions $C_1, C_2, \ldots$, form a multiply connected region.*

Proof: A path through $A \rightarrow C_1 \rightarrow B \rightarrow C_2 \rightarrow A$ cannot be contracted into single point.
QED

The algorithm, shown as function `topology` in Figure 9, works by searching for a decomposition of the set of c-regions into subregions such that the connectedness can be unambiguously determined using the 2 theorems above. When no such decomposition exists, the result must be multiply connected, as this is the only case where the decomposition can fail. The complexity of this procedure is significantly reduced by considering cliques of c-regions - which are known to be simply connected - as the elementary units of decomposition. In order to accurately determine the topology of the part which overlaps the initial environment E, it is important that the algorithm must only consider those connections which fall within E. This is achieved by the function `overlap`, which only returns the overlapping c-regions within E. Note that all operations can be implemented by subset tests on the maximal cliques of the region graph.

function `topology`(O,E)

1. $C \leftarrow$ maximal cliques within O.
2. for all $c \in C$ do
   (a) $e \leftarrow O \backslash c$, if e is empty return `simply-connected`
   (b) $o \leftarrow overlap(c, e, E)$, if o is empty return `not-connected`.
   (c) $tr \leftarrow topology(e, E), to \leftarrow topology(o, E)$
   (d) if `tr` = `simply-connected` and `to` = `simply-connected` return `simply-connected`
   (e) if `tr` = `simply-connected` and `to` = `not-connected` return `multiply-connected`
   (f) if `tr` = `not-connected` and `to` = `simply-connected` return `not-connected`
3. if no result has been found: return `multiply-connected`

function `overlap`(X,Y,E)
return a list of all c-regions $x \in X$, $y \in Y$ such that $\{x, y\} \cup E$ is a clique.

Figure 9: *Algorithm for computing the topology of the overlap set O of an environment E.*

5 Rotations

When the moving object is allowed to rotate, four important differences appear:

- the topology of c-regions includes the doubly connected rotation group S^1, and thus c-regions are doubly connected.
- intersections of c-regions can be multiply connected or consist of multiple subregions.
- The three-dimensional configuration space admits S^2 as another subgroup.
- Theorem 3 must now be applied to the 4-th order region graph (rather than 3-rd order).

These differences imply a number of complications to the algorithm, which however do not change its principal properties. The details of the modifications are more involved and beyond the scope of this paper. Implementation of the technique for the case of rotations is currently under way.

6 Implementation

I have implemented the techniques 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 compute the region graph for the configuration space by exhaustively searching for simultaneous overlaps. This is by far the slowest operation since it performs iterative approximations on bitmaps. The preprocessor defines the set of obstacles and bubbles, and a graphical interface permits visualizing sample configurations in each.

Because arcs in the region graph represent intersections between regions which are present in a set of configurations, there is a high probability of finding them by random generation. Computation of the region graph can be made much more efficient by first generating a number of random configurations, and noting the part overlaps they exhibit. Only simultaneous overlaps which have not been ruled out and which have not been found by this procedure need to be searched for explicitly.

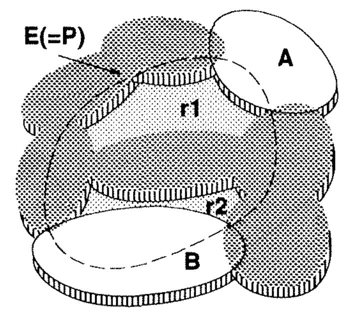

Figure 10: *Environment E (and consequently place P) models two disjoint regions, thus leading to incorrect connectivity between A and B.*

The implementation shows that it is indeed possible to perform spatial reasoning on the basis of bitmaps only. In practical applications, computations could first be carried out on polygonal approximations of objects, and representations of the precise shapes need only be used in case of ambiguities.

7 Discussion

Resolution limits of cavities The resolution of the free space representation is limited by the cavities covering it. In particular, a single environment E of bubbles, and thus a single place P, may cover several disjoint regions $r_1, r_2, ...$ of free space. The main problem caused by this phenomenon is that the connectivity of free space computed by the program may be incorrect by predicting a path from $A \leftrightarrow P \leftrightarrow B$ when the existing connections are in fact $A \leftrightarrow r_1$ and $B \leftrightarrow r_2$ with no connection between r_1 and r_2 (Figure 10).

Insufficient resolution can be detected in the topological computation by the fact the the overlap set of E will be more than doubly connected, i.e. contain several "holes". This can be detected by modifying the topology computation algorithm described earlier so that it can return a different default solution if the best decomposition found is one which implies several holes. This problem can only be solved by increasing the resolution of the cavities whenever it is insufficient. In our current implementation, we only signal the problem; increasing the resolution is up to the user.

Using prior knowledge If a place vocabulary has already been computed for a scenario involving some of the same objects, the corresponding parts of the region graph can be reused to solve a new problem. For example:

- when an obstacle is moved between two scenarios, the new region graph can be found by only recomputing those arcs which involved the moved obstacle. Only parts of the place vocabulary which depend on the newly created cliques need to be recomputed.
- the region graph for a situation could be composed of prototypes. A pattern recognition procedure could serve to identify the right prototypes to model each combination of obstacles.

The possibility of using prior knowledge makes it possible to envisage algorithms for computing place vocabularies for complex situations in real time.

Extension to qualitative kinematics Besides predicting the possible regions and their connectivity, it is often important to reason about the kinematics of contacts between surfaces. To do this, it would be useful to know the contact relationships which can be reached from a given place without passing through any other one. The region-based object representation is not detailed enough to represent contacts themselves, but only combinations of object pieces which could come into contact; these are represented as obstacles. For each place P, the overlap set of its underlying environment gives directly the set of obstacles which are adjacent to it, and might have to be considered for kinematic analysis.

Three dimensions The work presented in this paper is restricted to two dimensions, primarily to simplify graphical representation. For the case of pure translation, Theorems 2 and 3 generalize without modification since they only refer to the convexity of sets. I have not investigated the problem of three-dimensional rotations yet.

Robustness An important problem with all geometrical computation ([4]), and especially computation involving configuration spaces, is that small errors in the numerical computations can cause grossly erroneous results. For example, an algorithm which computes configuration spaces by tracing out its boundaries will give an entirely incorrect topology even if only one connection of the boundaries is computed incorrectly.

The topology-based computation is very robust against such errors. When a numerical errors results in predicting a single non-existant overlap, this normally creates an additional clique containing only that overlap. Since the clique will be very small, it will most likely not be a superset of any interesting environment, and thus have no influence on the place vocabulary. When an overlap is missing, this can result in a clique being broken into two smaller subsets. In the case where this overlap is indeed the only one ruling out a place, this will cause a spurious place to appear, but not affect the rest of the place vocabulary.

Complexity Since each maximal clique models a feasible region of configuration space, its number does not grow more than $O(n^d)$, where n is the number of c-regions and d is the dimensionality of the configuration space. Since in a search algorithm all leaf nodes are maximal cliques with one more c-region added, the complexity of finding them is no greater than $O(n^{d+1})$.

The second important part of the algorithm is to compute the topology of region overlaps. This involves a search which is in fact of exponential complexity, but the number of c-regions which can overlap any environment is limited by the fact that the moving object cannot overlap parts which are farther apart than its size would allow. Thus, as the number of c-regions gets large, the complexity of determining the overlap topology of an environment is about constant. The number of overlap region computations is bounded by the number of environments which are examined. There are not more environments than feasible regions of configuration space, and thus their number again does not grow by more than $O(n^d)$.

Thus, the total complexity of the algorithm can be estimated to be about $O(n^{d+1})$. In practice, the computation is very fast: for the example of Figure 1, it takes less than 1 second to compute the place vocabulary on the basis of the maximal cliques.

8 Conclusions

The novelty of the spatial reasoning technique presented in this paper lies in two aspects: using a region-based object representation, and using topological rather than geometrical properties for computing a place vocabulary.

Because the method is based on topology, it completely avoids the problems associated with algebraic surface models:

- object representations as unions of convex parts have long been postulated in vision research ([5]), and there are reliable methods for computing them.
- the methods are robust: inaccuracies in the possible overlaps have only local influence on the place vocabulary. Furthermore, since any overlap of c-regions always consists of a *set* of points, the numerical analysis required to find them is simpler than in cases where precise configurations must be detected.
- information about simultaneous overlaps obtained from previously solved subproblems can be directly reused in other contexts. Furthermore, it is possible to use *abstractions*: groups of parts can be approximated by their convex hull until a more precise representation is needed.

However, the technique requires that objects can be represented as unions of convex parts and thus cannot deal with concave surfaces. Furthermore, it requires a method for deciding the existence of configurations where certain combinations of small sets of parts and cavities can simultaneously overlap.

Further development of the methods for three-dimensional problems, rotations and multiple moving objects are currently under way. I expect it to yield much simpler methods for spatial reasoning than were possible with geometric boundary-based methods.

References

[1] **V. Chvátal**:"Linear Programming," W. H. Freeman, 1983

[2] **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*, San Jose, July 1992

[3] **Boi Faltings**: "Qualitative Kinematics in Mechanisms," *Artificial Intellgence* 44(1), June 1990.

[4] **C.M. Hoffman**: *Geometric and Solid Modelling: An Introduction*, Morgan-Kauffman Publishers, 1989

[5] **D.D. Hoffman, W.A. Richards**: "Parts of Recognition," *Cognition* **18**, 1985

[6] **Y.K. Hwang, N. Ahuja**: "Gross Motion Planning - A Survey," *ACM Computing Surveys* **24**(3), 1992

[7] **William S. Massey**: "A Basic Course in Algebraic Topology," *Graduate Texts in Mathematics*, Springer, 1991

[8] **E. Spanier**: "Algebraic Topology", Mc. Graw Hill, 1966

[9] **J. T. Schwartz, M. Sharir**: "On the Piano Movers Problem. II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds," Advances in Applied Mathematics,4, 1983

[10] **J.T. Schwartz, C.K. Yap**: *Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics*, Erlbaum, Hillsdale, N.J., 1987

[11] **T. Lozano-Perez, M. Wesley**: "An Algorithm for Planning Collision-Free Paths Among Polyhedral Obstacles," Comm. of the ACM, **22**, 1979,