Applying Means-ends Analysis to Spatial Planning

Boi Faltings
Laboratoire d'Intelligence Artificielle (DI)
Ecole Polytechnique Fédérale de Lausanne
MA - Ecublens, 1015 Lausanne, Switzerland

Pearl Pu
Department of Computer Science
University of Connecticut
U-155, Storrs, CT 06269, USA

January 22, 1992

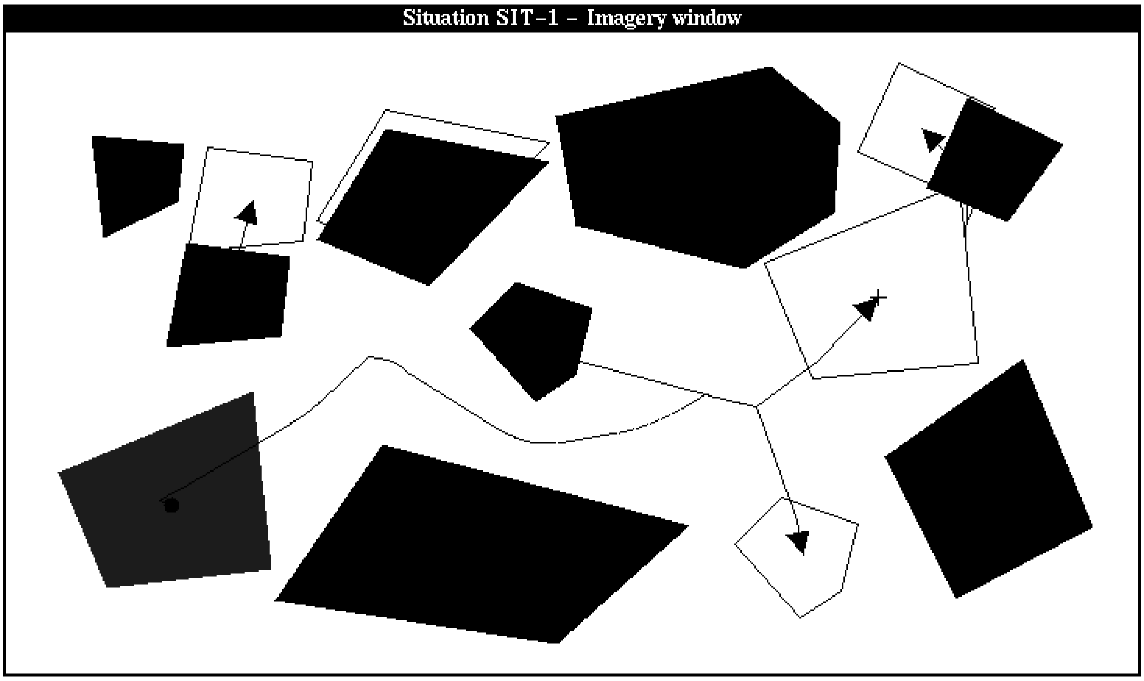

Figure 1: *An example of a spatial planning problem. The goal is to move the object shown in black to the goal position indicated by a cross. This requires moving some of the obstacles, shown in grey. The motions of the plan found by the planner are indicated by arrows drawn along the corresponding Voronoi edges, and the final positions by hollow shapes.*

Abstract[1]

Currently known methods for robot planning fall far behind human capabilities: they require approximations of shapes, and they cannot generate plans which involve moving obstacles to clear a path for the moving object. In this paper, we explore the hypothesis that *means-ends* analysis based on a world model involving *mental imagery* allows more human-like solutions. Our method is based on a novel way of representing planning constraints which makes it possible to incrementally generate the symbolic representations for means-ends planning using only imagery operations.

[1]A similar paper appears in the Proceedings of the International Workshop on Intelligent Robots and Systems 1991, IEEE Press.

1 Spatial Planning Problems

The problem of planning a sequence of actions to move objects into a desired configuration is fundamental to intelligent robotics. Path planning algorithms have reached a sophistication where they even exceed human performance, but only as long as the goal can be achieved by moving only a *single* object. In this paper, we address the *spatial planning problem*, which we define as the problem of moving an object in a space containing an unbounded number of *movable* obstacles. We are particularly interested in examples such as the one shown in Figure 1, which also requires moving some of the *obstacles* to clear a path for the moving object.

Although there exist complete algorithms which can solve such problems (see for example [3]), it is known that a restricted form of the spatial planning problem, the *warehouseman's problem*, is PSPACE-complete ([6]). Under the assumption that P $\neq$ NP, it follows that any provably *complete* algorithm for spatial planning will not be tractable for realistic situations involving a large number of movable obstacles. On the other hand, most practical examples have very simple solutions, and it is not clear that a motion planner has to be *complete* to be practical.

People solve spatial planning problems routinely and seemingly without great effort. Even more surprisingly, their performance does not seem to depend at all on the number of distinct obstacles present in the environment. This contradicts current approaches to spatial planning, where the consideration of potential motions of all individual obstacles entails a strong dependence of the runtime on the number of movable objects. Big differences exist also in the performance characteristics of the human spatial planning "algorithm": it is more *efficient* with a broader *coverage* of problems, but it is probably *incomplete*.

The goal of our research is to develop spatial planning algorithms with characteristics similar to people. We believe that for many applications, such as autonomous robots, the advantages of coverage and efficiency far outweigh the problem of incompleteness. Our research philosophy is therefore not to develop complete algorithms, but to find those methods which are *efficient* and *powerful* and characterize their incompleteness.

Psychological analysis ([12]) proposes *means-ends analysis* as a theory of how people deal efficiently with the unbounded possibilities of choice involved in spatial planning. In means-ends analysis, operators are only considered if they are relevant to the goals of the system. The efficiency of such a system is not affected by the number of operators, because most of them will never be used.

Since means-ends analysis links a discrete set of goals to a discrete set of operators which can achieve them, it requires a *symbolic* planning framework. However, spatial planning involves object shapes and positions which are inherently numerical. The main issue in applying means-ends analysis to spatial reasoning is the *qualitative* representation of space for use by the planner. There exist methods for computing this representation a priori, such as in the path planning methods based on configuration space ([10, 14]),and qualitative spatial reasoning based on place vocabularies ([5]). Computing the qualitative representation independently of its use means including *all imaginable* possibilities, and contradicts the basic principle of means-ends analysis. Furthermore, we shall see that if the representation is to be sufficiently expressive for symbolic planning it would amount to an exact numerical representation.

Computing qualitative representations of space requires heavy numerical computation; carrying it out during the planning process brings up problems of *efficiency*. Another clue which psychology gives us is that people seem to rely heavily on *mental imagery* for solving spatial problems ([9]). Planning with subgoals implies the use of backward chaining, which means that situations are often incompletely specified. Up to now, it has been hard to see how imagery could represent such situations. An important result of this paper is the discovery that imagery can be used to represent conditions and partial descriptions in the same framework as objects themselves. In this way, it can incrementally generate the qualitative representations required for the planner based only on operations which could be carried out in constant or linear time on massively parallel processors.

Based on this discovery, we have constructed a simple prototype which demonstrates the use of means-ends analysis for the simple domain of planar objects without rotation. In spite of its simplicity, our planner is capable of solving problems like that of Figure 1 and should encourage further research for applying the approach to practical problems.

2 Situation Calculus for Space

Means-ends analysis was first described in the psychological models of Newell and Simon which resulted in the General Problem Solver ([12]). It was applied to robot planning in the work on STRIPS ([4]) and is the basis of a large body of work on planning systems ([1]). The basis for all means-ends planning is the *situation calculus* representation, which defines a symbolic world model on which the planner operates, and the main issue of this paper is how to formulate this model.

Our situation calculus defines the following elements:

- **Situations** are states of affairs at a particular time, and their representation is often only a partial description. A spatial situation is an arrangement of objects, and a partial description of an object position is expressed by a set of constraints.
- **Actions** transform a situation S into a new situation S'. In spatial planning, actions are motions of objects to qualitatively different positions. Actions are generated specifically for achieving particular goals.
- For each action, a set of **preconditions** which must hold in a situation for the action to be applicable. In our case, the precondition for a motion is that there exists a collision-free path for carrying out the motion.
- **Goals** are the motivation for actions. In our case, goals are that a *moving object* should be moved from its *originating position* to the *goal position*, both of which are precisely specified. **Subgoals** follow the same format, but refer to moving obstacles to positions where they no longer interfere with the precondition for an action.
- **Protections** are constraints which prevent actions from clobbering the preconditions for subsequent ones. A protection mechanism is important in spatial planning, since an obstacle affects the possible motions of *all* other objects.

The partial descriptions involved in planning are handled most efficiently by using a single qualitative representation of space throughout the entire planning process. We now show that such a qualitative model might have to be infinitely precise, and thus amount to an exact numerical model.

A qualitative representation of object positions in a finite system defines a *place vocabulary* ([5]), which is a

Figure 2: *Qualitative representations of motions. In situation S, there are qualitatively different motions Ax_1, Ax_2 and Ax_3 for object X. In preceding situations, the space required for the motions must be kept clear, and the qualitative motions for the object Y in situations S' (to be followed by motion Ax_1) are different from those in S'' (to be followed by Ax_2). S_i and S_k are other situations which might precede S or S'.*

decomposition of the space into a finite set of regions. The minimum criterion for the usability of the representation is that it must be sufficiently expressive to allow the planner to verify the preconditions for its actions. However, it turns out that:

> Representing situations with sufficient precision to evaluate preconditions in a sequence of actions of any length requires a numerically precise representation of positions.

This result follows from the following argument. Consider the possible motions of two objects, X and Y. Assume that in some situation S, there are n qualitatively different regions to which object X can be moved (Figure 2).

Now consider situations S', S",..., preceding S, in which object Y has to be moved, and which must be qualitatively representable in a single framework Q. For each of the n regions in S, there are at least 2 regions in Q: one where Y would block the subsequent motion of X, and one where it would not. As the actual combination of motions is not known a priori, Q must contain all *combinations* of these regions. Under the assumption that there are no singularities, this results in at least $O(n^2)$ regions for the placement of Y. By induction, a representation which would correctly predict the preconditions for a sequence of k different object movements would distinguish $O(n^k)$ qualitatively different positions for motion of the first object in the sequence. If k is unbounded, this representation becomes infinitely fine and approaches a numerical representation.

A similar result has been observed by Struss & Hubermann ([7]) for the qualitative analysis of certain types of billard games. It means that a finite qualitative representation *by itself* is insufficient for planning.

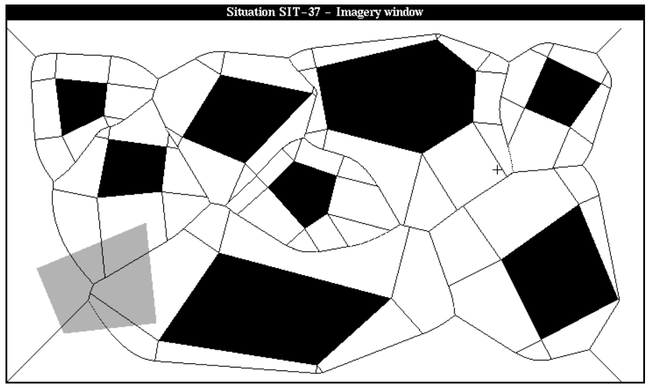

Figure 3: *A Voronoi diagram represents all topologically distinct paths in a finite graph. The moving object, in the lower left corner, is not considered in the computation.*

3 Situation calculus with imagery

An alternative solution is to compute the required qualitative models specifically for the requirements of the planner. This is the idea of *imagery*: interpreting a precise representation in qualitative terms according to the needs of the current problem solving requirements.

In the resulting paradigm, qualitative representations of possible motions are constructed incrementally by backward chaining, starting with the desired goal state. In the example of Figure 2, different qualitative representations are computed for S', S", ..., depending on the actions which are to follow. Motions to the final goal are represented as *regions* which constrain the qualitative representations for earlier plan steps, shown shaded in Figure 2. The planner only generates those region structures which are actually necessary, and avoids the multiplication of possibilities at each step.

More precisely, the situation calculus model is simulated by the following numerical computations, which define the issues to be addressed:

- there is an explicit numerical representation of each situation. This raises the problem of how to represent incompletely specified situations.
- at each stage, a numerical computation determines the set of qualitatively different object motions and their preconditions. The problem here is how to finitely represent all possible motions.
- for each violated precondition, subgoals are computed using the representation of the situation. This requires reasoning about the actions required to clear a path.

Voronoi diagrams The *Voronoi* diagram of an arrangement of polygons (Figure 3) is defined as the set of all points which have the same distance from two polygon boundaries, and plays a fundamental role in our representation. The Voronoi diagram is a division of space into regions defined by the criterion that all points of each region are closest to one and the same boundary. It is also a graph which represents all topologically distinct motions of an object, such as the polygon on the lower left corner in

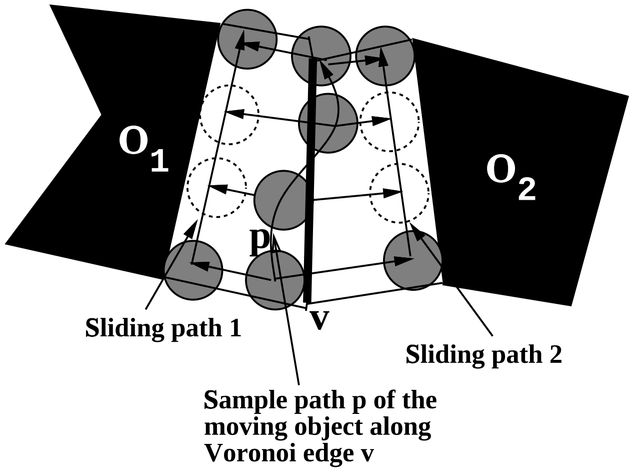

Figure 4: *Any path of the moving disc corresponding to the Voronoi edge between the two black objects can be mapped continuously to one of the sliding paths.*

Figure 3. Voronoi diagrams are an important link between numerical and symbolic representations.

For polygonal obstacles, the Voronoi diagram can be computed by efficient recursive algorithms ([8]). Objects with arbitrary shapes can be represented as bitmaps whose Voronoi diagram can be computed using the *medial axis transform* ([11]).

Representing incompletely specified situations In spatial planning, the only information that a goal gives about the desired final situation is the final position of the moving object, resulting in considerable uncertainty about the positions of the other objects. As the planner proceeds backwards in time, the actions it plans impose more and more constraints on the intermediate situations: they must permit the sequence of actions that follows them in the plan.

The constraints implement *protections* ([15]) of the preconditions of later actions: obstacles may not be placed in positions where they would block a later action. It would be straightforward to define protections if the actions involved *particular* paths: nothing may interfere with the region that is traced out by the motion. However, exact paths are unknown and we must protect *necessary* conditions for the existence of *any* path. The following remarkable property of the Voronoi diagram formalism makes it possible to define a single *particular* motion whose existence is equivalent to the *necessary* condition for the existence of *any* motion of a certain type.

A path p along a Voronoi edge v between obstacles O_1 and O_2 has the property that all positions on p fall within the Voronoi cells V_1 and V_2 and thus have O_1 or O_2 as the closest obstacle. This implies that if in any position the moving object overlaps obstacles, one of the overlapping obstacles must be O_1 or O_2[2]. Conversely, if a position on p has no overlap with O_1 or O_2, we can conclude that it is legal.

[2]This is strictly valid only if the moving object is a disc, but we shall see that it simply results in an overly conservative protection if this condition is not satisfied.

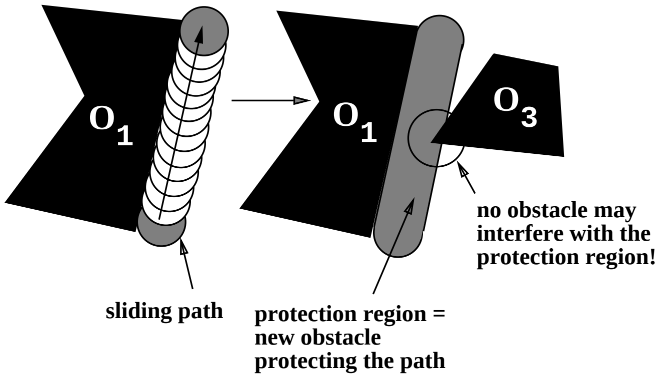

Figure 5: *The feasibility of a sliding path can be* protected *by creating an imaginary obstacle corresponding to the area swept out by the object moving along this path.*

Assume now that there is a legal path p along v. As shown in Figure 4, p can be continuously transformed into one of two *sliding paths* where the object is moved as close as possible to the obstacle boundaries. By convexity, if p was already legal, this tranformation cannot create any new overlaps with either O_1 or O_2, and consequently with other obstacles. Therefore, p is legal only if both sliding paths are legal.

Figure 5 shows how the sliding paths can be exploited to generate protections for a path p. Assume that an obstacle O_3 is moved close to the region between O_1 and O_2. If a path p along v is preserved by this motion, p must pass either between O_3 and O_1, or between O_3 and O_2. It remains feasible if and only if the corresponding *sliding* path remains feasible. Thus, the transformation preserves a possible path along v if it leaves at least one of the sliding paths feasible. In our implementation, for each Voronoi edge v the planner chooses the sliding path which is to be protected.

We have thus expressed a necessary condition for the existence of *any* path as a the condition for the existence of one *particular* path. This defines a *protection region* of space into which no other obstacle may be moved, as shown in Figure 5. All obstacles are subject to the same constraints: they may not interfere with any protection region for a later motion. We emphasize that the planner makes an explicit choice between two different versions of each topological motion, and backtracks on this choice if no solution is found.

Note that this technique is only valid if the shape of the moving object is a disc, and the condition is too strong for other shapes. Furthermore, the Voronoi diagram and protection regions have to be recomputed after every obstacle movement, for otherwise there could be paths whose protection is not covered by the sliding paths. Ignoring these conditions, as our implementation does, means that the planner may miss certain paths, but not that it can generate nonexisting and thus erroneous paths. It is thus not a serious practical problem.

In order for motions to follow one another, the moving object must sometimes change the obstacle it is sliding on. This defines a third type of path, the *crossover* path. The protection region for a crossover path is chosen so that the object slides along one object for half the way, then crosses over to the other object and continues its motion there. The exact shape is arbitrary, and another source of planner incompleteness.

In order to choose the optimal paths to plan in a situation S, the planner makes the default assumption that obstacles remain in their original position. This is reasonable since normally only a small part of the obstacles are moved. The fact that some obstacles will have been moved to satisfy subgoals of earlier actions may make these actions suboptimal, and this is the only source of incompleteness which has turned out to be practically significant. The advantage of making this assumption is that the planner can use one and the same Voronoi diagram throughout the planning process. It could be avoided by replanning the actions up to the end of the plan whenever an obstacle is moved. This makes the search algorithm more complex and we have not implemented this solution.

Actions and their preconditions Any path can be continuously transformed to a sequence of Voronoi edges by a *retraction* ([13]) mapping. In particular, retraction associates the precise goal positions with arcs of the Voronoi graph.

The set of possible actions is given as the motions along the Voronoi edges which are adjacent to the current position of the moving object. Each motion exists in four different versions, corresponding to the two extremal paths and the crossovers between them. The *preconditions* for these versions are defined as the non-interference of obstacles with the protection region of the corresponding extremal path. Whenever an obstacle interferes with a protection region, the planner generates a subgoal to remove the obstacle to a non-obstructing position.

Our current planner uses *one and the same Voronoi diagram throughout the planning process*, namely the Voronoi diagram of the initial situation. This means that only paths which topologically already existed in the initial situation can be considered by the planner, and is a source of planning incompleteness. However, it is hard to imagine examples where this would actually be a problem.

Defining subgoals A path is passable if and only if nothing interferes with its protection region. The minimal condition for making a path passable is to remove offending obstacle outside of the protection region. At the same time, the obstacle must not clobber the precondition for a subsequent motion. This defines a qualitative region of possible positions to which the obstacle can be removed, which is explicitly computed using a *configuration space* transformation ([10]). Note that this definition of subgoals means that the planner cannot generate plans where *both* obstacles bounding a path have to be moved, another source of incompleteness.

The system picks the Voronoi nodes which fall inside the legal regions as goal positions, and solves the subgoals in parallel to find the least costly solution. Note that the exact choice of position is not important, since all points within the same simply connected region are qualitatively equivalent with respect to the planning constraints. Only when several obstacles are removed simultaneously, their positions may conflict with one another. We can use a generalized version of constraint propagation ([2]) to avoid conflicting positions: the region of legal positions for moving each object is iteratively decreased by the subregions where it would leave no solution for moving one of the others.

Functions of the imagery module An important aspect of our system is that all numerical computation are carried out by an *imagery module*, ideally operating on bitmaps using special massively parallel hardware. The imagery module must be able to perform the following operations, which we are currently simulating using techniques of computational geometry:

- **Basic imagery operations**: The module can perform the basic bitmap operations of region composition and complement, and finding the boundary of a region by tracing.
- **Voronoi diagram computation**: The *Medial axis transform* of the image of the obstacles results in the Voronoi diagram, which is passed to the symbolic module. The imagery module labels each edge of the Voronoi diagram with the obstacles it separates, an information which is obtained by colouring techniques in the medial axis transform algorithm.
- **Generating protection regions**: Protection regions are generated by projecting the endpoints of the corresponding Voronoi edge on the object boundary, and simulating the slide of the moving object along the edge between these points. The protection region is the region swept by the object in this process.
- **Configuration space transformation**: Growing obstacles by the size of a moving object creates a space which represents the set of positions where no overlap would exist between objects. This is carried out by a simulation of the moving object sliding around all boundaries of the obstacles, and composing the corresponding regions.

4 Conclusions

We have shown a new approach to spatial planning inspired by human performance. It has significantly broader *coverage* and *efficiency* than any currently known motion planner, but this is achieved at the price of *incompleteness*. Our method is based on a new way of formulating necessary conditions for the existence of *any* motion as conditions for a *particular* motion, which allows representing these constraints in imagery. While the methods are *incomplete*, they are *sound* and never generate plans that are in reality impossible.

We have implemented the methods discussed in this paper in a prototype system for the simple domain of two-dimensional, polygonal objects without rotation. The prototype has performed very well, and we do not know of any other program which can efficiently generate plans of the complexity shown in Figure 1. Even though our approach is incomplete, we consider the initial results promising enough to encourage further research.

We believe that the combination of imagery and symbolic reasoning which our planner exhibits is a promising methodology for spatial planning, since:

- the symbolic framework of means-ends planning allows *combinatorics* and can make *choices* between alternative plans. Explicit choice is necessary to avoid problems inherent in planning based on imagery-like operations, for example the local minima problem in potential field planners.
- means-ends analysis makes it possible to design an algorithm whose runtime depends on the complexity of the solution, but not on the number of potential solutions.
- the numerical framework of mental imagery makes paths and interactions emerge from general shape representations and thus deals with the *representational* problems of spatial reasoning.

The current prototype is only intended to explore the feasibility of the concepts and suffers from several weaknesses. First, the program is not optimized and therefore rather slow, and many remaining bugs make its use difficult. Second, the planner uses a plain A* algorithm with very simple heuristics, which is quite inefficient. Third, we do not have access to a massively parallel computer and have restricted the shapes to polygons so that the imagery operations can be efficiently simulated by computational geometry. We hope to extend the current prototype in order to better characterize the ways in which our method is incomplete. In the long term, we would like to investigate if similar methods are possible in more general domains, allowing three dimensions and object rotations. We also think that the use of imagery offers new possibilities for future qualitative reasoning systems.

References

[1] **J. Allen, J. Hendler, A. Tate**: "Readings in Planning," Morgan-Kaufmann Publishers, 1990

[2] **E. Davis**: "Constraint Propagation with Interval Labels," *Artificial Intelligence* **32**, 1987

[3] **M. Erdmann, T. Lozano-Perez**: "On Multiple Moving Objects," *Algorithmica* **2**(1987)

[4] **R. E. Fikes, N.J. Nilsson**: "STRIPS: A new Approach to the Application of Theorem Proving to Problem Solving," Artificial Intelligence **2**, 1971

[5] **K. Forbus, P. Nielsen, B. Faltings**: "Qualitative Spatial Reasoning: A Framework," to appear in *Artificial Intelligence*, 1991

[6] **J. Hopcroft, J. T. Schwartz, M. Sharir**: "On the complexity of motion planning for multiply independent objects: PSPACE-hardness of the warehouseman's problem," TR 103, Courant Institute of Mathematical Sciences, NYU, 1984

[7] **B. Huberman, P. Struss**: "Chaos, Qualitative Reasoning and the Predictability Problem," *Recent Advances in Qualitative Physics*, MIT Press, 1991

[8] **D. Kirkpatrick**: "Efficient computation of continuous skeletons", Proceedings of the 20th FOCS, 1979

[9] **S. M. Kosslyn, S. Pinker, G. E. Smith, S. P. Shwartz**: "On the Demystification of Mental Imagery," in: Ned Block (ed.): Imagery, MIT Press, 1981

[10] **T. Lozano-Perez, M. Wesley**: "An Algorithm for Planning Collision-free Paths among Polyhedral Obstacles," Communications of the ACM **22**(10), October 1979

[11] **U. Montanari**: "Continous Skeletons from Digitized Images," Journal of the ACM 16(4), Oct. 1969, pp. 534-549

[12] **A. Newell, H. Simon**: "GPS, A Program that Simulates Human Thought," in E.A. Feigenbaum & J. Feldman (eds.): *Computers and Thought*, Oldenbourg, 1963

[13] **Colm Ó'Dúnlaing, Chee K. Yap**: "A Retraction Method for Planning the Motion of a Disc," in *Planning, Geometry and Complexity of Robot Motion*, Ablex Publishing, 1987

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

[15] **Richard Waldinger**: "Achieving Several Goals Simultaneously," SRI AI Center Technical Note 107, SRI, Menlo Park