Proceedings of the IEEE Regional Conference on Aerospace Control Systems, Westlake Village, CA, May 1993.

Nested Motion Planning for an Autonomous Robot

J. Albus*, A. Meystel, S. Uzzaman

Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104
***Robot Systems Division, National Institute of Standards and Technology, Gaithersburg, MD 20899**

Abstract
This paper addresses some of the issues associated with the planning in hierarchical systems. Specifically, it is required that controls for an autonomous vehicle be synthesized as the result of the nested hierarchical analysis of the minimum time motion problem. It is demonstrated that search in the output space allows for efficient planning/control procedures. The results of 2-level nested hierarchical planning are given as a representative simulation example.

***Key words:** Autonomous Robots, Motion Planning, Hierarchical System, Low and High Resolution, Maneuver Planning and Search.*

1. Introduction

The method of consecutive refinement is a strategy of problem solving which is utilized extensively in human decision making. The essential element of this strategy is the ability to construct world models which constitute a hierarchy of generalization from the highly abstract down to the most inclusive and exact. These models then become the source of guidance in the synthesis of successively more detailed and exact plans or designs and, in the domain of control systems, the implementation of dynamic controls [1, 2].

Our approach to robot motion planning is based upon a multiresolutional world representation with consecutive refinement of plans, and with the eventual automation of a rule based guidance system for the selection of strategies of stratification (e.g. see materials on NASREM which are broadly applied in a multiplicity of different projects and described elsewhere). It is shown in a number of sources, for example [3], that this hierarchical organization of information allows for analysis with reduced complexity because it can be arranged that the space under consideration at any given level of the hierarchy contains only a small fraction of the information related to the overall problem.

Probably the most significant issue in this respect, is that of consistency of the hierarchical partitioning of modeling information; although this is still a subject of research, we suggest a measure of consistency in this paper and show by example that the hierarchical organization of information can transform search-based planning for control of dynamical systems into a solvable form.

In this paper we would like to emphasize that if one applies this approach consistently, it can be a basis for a universal solution equally applicable at each level of resolution. Indeed, after the general motion is planned, and one is interested in a sequence of driving commands, most of the sources recommend to switch to the rule based system and fill in the table of rules heuristically and/or based upon expert knowledge. We will show that when the resolution grows and the vehicle becomes a four-wheel machine instead of a dimensionless point, the solution can remain the same and no expert rule control is required.

2. The Hierarchical Generalization of System Models

In this section we outline the premises of the algorithms recommended. Levels of generalization and hierarchical representation, as discussed here, are considered to be depictions of the same object with different degrees of accuracy. We will formalize the preceding statement in mathematical form by applying concepts of the usual state space description for the (not necessarily linear time invariant) system:

$$\dot{x}(t) = A(x,u,t)x(t) + B(x,u,t)u(t)$$
$$y(t) = C(x,u,t)x(t)$$

where

$$x \in R^n, \ u \in R^m, \ y \in R^p, \ t \in R^+$$

Thus it is possible to form a solution of these equations as mappings describing the state transition and output functions:

$$\Phi : R^n \times R^m \times R^+ \rightarrow R^n \times R^+$$
$$\Psi : R^n \times R^m \times R^+ \rightarrow R^p \times R^+$$

so that for any input function "u" on the interval [t_0, t_f] it is possible to determine the corresponding output function "y" on the same interval. If it can be shown that there exists a pair of functions

$$\Phi' : R^{n'} \times R^{m'} \times R^{+} \rightarrow R^{n'} \times R^{+}$$
$$\Psi' : R^{n'} \times R^{m'} \times R^{+} \rightarrow R^{p'} \times R^{+}$$

for which n' is strictly less than n, and for which the same input function "u" generates the output function "y'" such that inequality

$$\left\| \int_{t_0}^{t_f} (y'(t) - y(t))dt \right\| < \varepsilon$$

holds for all admissible inputs in the input function space where ε is a value which depends on the level of resolution under consideration. Then, it is claimed that

$$\{\Phi', \Psi'\} \text{ is an } \varepsilon \text{ - generalization of } \{\Phi, \Psi\}.$$

The strictness of this formulation may be relaxed by considering a stochastic measure for associating a confidence level with the generalization to construct the concept of ε – generalization *nearly everywhere*. Thus,

$$P[\left\| \int_{t_0}^{t_f} (y'(t) - y(t))dt \right\| < \varepsilon] < \tau$$

is a statement of the belief that the constraint holds with a probability defined by the preassigned threshold τ.

This formulation can be extended to an ordered collection of epsilons, $\{\varepsilon_1, \varepsilon_2, \ldots , \varepsilon_k\}$, thereby defining a hierarchy of models which describe the same input-output behavior with increasing degrees of accuracy. The necessity of considering all elements of the input and output vectors as time varying functions may also be relaxed so that at some level 'i', $u_{k_i}[t_0, t_f]$ could be considered constant in the interval, whereas at some lower level (at higher resolution) the same input may be represented as a time varying function.

The ability to formulate the world models with this hierarchical generalization will be shown in the following example to be an essential device for coping with the complexity associated with the planning of system operation in a combined feedforward-feedback controller.

3. Case Study: A Pilot for an Autonomous Vehicle

In order to begin planning the path of a vehicle in a space with obstacles, it is necessary to have a geometrical representation of the extent and the elements of the search space, including the vehicle. This job is accomplished by creating geometrical models of the workspace, vehicle, and obstacles. Templates for these objects are components of the simulation package SIMNEST developed by Drexel University with NIST participation for design and control of hierarchical systems. The package allows the customization of the workspace to accommodate new maps and vehicle parameters. However, the principles of representation and control are easy to apply in any simulation package.

Having constructed the iconic representation of the physical elements of the problem, the next step is to specify their behavior and interactions. In our case, a kinematic and dynamic model of the vehicle will be required, and it will be necessary to be able to identify the occurrences of possible collisions between the vehicle and obstacles so as to successfully avoid them.

The state of the vehicle in Figure 1 is often described in the literature in terms of the following variables: 1] x - position, 2] y - position, 3] orientation, 4] speed, 5] steering angle. We will consider this to be the level of 'complete informedness' or maximum resolution (even though we can include more complex issues such as the effect of roll on cornering dynamics or skidding) because it is sufficient to illustrate the principle of *successive generalization* required for obtaining a representation which fits into our concept of planning/control algorithm of *consecutive refinement.* (In addition it is necessary to know whether the vehicle is rolling or sliding so that, depending on the case, velocity can be considered to be a scalar).

Depending upon the acceleration imparted to the vehicle by the propulsion and steering actuators, it is possible to formulate a model of the system in which the next position of the car, given its current state, is a function of the effective steering angle and velocity, 'α' and 'v', during the interval of modeling.

The center of rotation of the vehicle is given by the point of intersection of the lines shown in Figure 1, corresponding to the extensions of the lines joining the fixed rear axle and the perpendiculars to the planes of the front wheels. (It is clear that the single steering angle is an abstraction of sorts because, for these three lines to meet at a point it is necessary that the two front wheels be rotated through different angles).

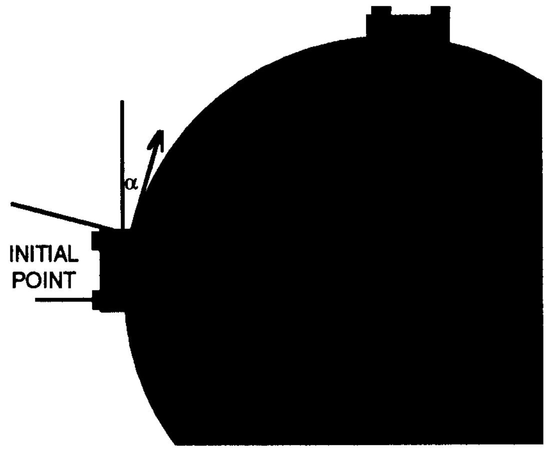

Figure 1. The construction of a system model for a 4-wheeled vehicle.

Since the distance 'd' moved on the circumference of the circles shown in Figure 2 can be determined from velocity and elapsed time to be

$$d = \int_{t_i}^{t_f} v \, dt$$

therefore the new coordinates of the vehicle with respect to the global xy position can be determined by solving simultaneously the equation of the circles

[i] centered at the vehicle with radius 'l' as shown, and

[ii] centered at the center of rotation with radius 'r'.

The four solutions of the quadratic equations

$$(x - x_1)^2 + (y - y_1)^2 = r^2$$

$$(x - x_2)^2 + (y - y_2)^2 = l^2$$

(where the subscripts refer to the indices of the centers of the two circles) correspond to the case of forward and reverse motion with positive (right turn) and negative (left turn) steer as shown in the figure. The correct global coordinates after moving the vehicle may be restored from the context in which motion occurs.

The orientation of the vehicle after motion occurs can be determined from the slope of the line joining the coordinates of the rear wheels of the vehicle, since this line is normal to the direction in which the vehicle is facing.

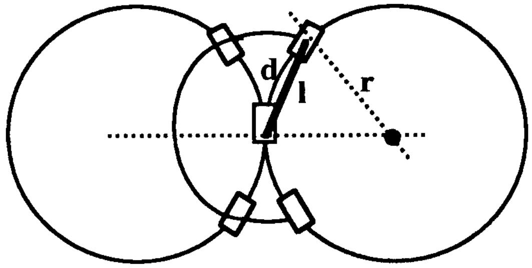

Figure 2. Position Computation.

The approach described is used to model the kinematics of the vehicle; the dynamics are a function of the actuators used in the vehicle.

Without loss of generality, the dynamics of the rolling vehicle can be modeled by the motion of an object with inertia subject to accelerating and braking forces. This model of motion is valid as long as (1) the vehicle does not slide because the force normal to its trajectory exceeds the reaction due to static friction, and (2) the wheels of the vehicle do not lose traction.

Assuming that other models exist for describing these regimes of operation, we can proceed with the completion of our 'comprehensive' model. The approach to system modeling at this stage, begins to come under the influenced of the desire to solve the motion planning problem using the principles of consecutive refinement. This means that the vehicle should be represented at more than one level of abstraction, and a rough plan should be created with an imprecise model before an attempt is made to synthesize the final plan. Since this example illustrates the use of two nested levels of planning, it will only be necessary to construct these two versions of the vehicle model.

4. The First Level: Low Resolution Planning (Trajectory Planning)

The technique of planning which is being exemplified here involves top-down search with successive refinement. It is assumed that the results of this search will be considered open loop control input for a particular level (a reference curve) with a feedback tracking controller compensating for current errors that could not be taken care of at the stage of planning. Thus, not only is it necessary to have the physical elements of the planning paradigm, a tool for performing the search must also be available. This component is described in [3]. It is also available as one of the options in the procedures library of SIMNEST. In this procedure, in order to perform the search, a graph representation of the workspace must be developed, an operation which is fairly complex as a result of the desire to incorporate successively more refined information in the two descriptions to be used for search.

In generating the graph of the first level, the model that is used is one that allows the inclusion of the least amount of detail that is considered sufficient to provide rough predictions of qualitative dynamical behavior. This is accomplished by representing the vehicle as an object whose velocity is conditioned by its change of direction. Thus if the base velocity of the object is v_{max}, then the velocity during motion requiring a change of direction of θ degrees is:

$$v = v_{max} \cos \theta$$

In this rudimentary model, therefore, only the resistance to change of direction is shown; other kinematic or dynamic issues are not represented. In order to utilize this model for planning, the first level search procedure begins by discretizing the horizontal workspace of the vehicle. The rule of discretization is novel, in that a regular grid is not utilized. Instead, the xy plane is partitioned into regular subsets, and a random coordinate pair is used to represent each subset. This approach eliminates the idiosyncrasies associated with regular grids without utilizing a biasing heuristic. The number of points used to represent the space is a function of the average inter-point spacing and the narrowest passageways in the map. The influence of these parameters can be determined by experiment.

The connectivity of the randomly discretized version of the workspace is decided by the definition of vicinity at a given level of the representation. If the concept of vicinity is relaxed to include the whole space, the graph could hypothetically be fully connected. On the other hand, curtailing the measure of vicinity may lead to a graph which is not connected. This problem can be addressed by using heuristics, or by trial and error. There remains the issue of whether edges in this graph traverse forbidden regions in the workspace or whether the vehicle will intersect obstacles during such traversals. Such potential intersections are tested individually during the construction of the graph. The concept of the vehicle being represented as an expanded point is exercised and the degree of expansion is fixed by iterations beginning with the minimum dimensions of the vehicle.

The search minimizes the estimated time of traversal at each level of representation. At the first level of resolution, edge cost is computed using the expression:

$$t = \frac{d}{v_{max} \cos \theta}$$

where t is time (cost), d is the Euclidean distance between neighboring points, v_{max} is the maximum velocity of the vehicle and θ is the change in direction between consecutive points on the trajectories being tested. Thus, the point to point traversal is assumed to occur at maximum velocity unless it involves a change in direction, which slows down the vehicle.

Upon completion of the graph representation of the workspace, a version of the best first graph search algorithm (the so called best-first algorithm) is invoked to determine a path between the start and the goal positions in the graph. The algorithm is not exactly the same as that described in [4] because the graph may conceivably be generated during search, the result of search is not necessarily (or even probably) globally optimal and it is being applied to a different class of problems. However, the search process is at least similar to Dijkstra's algorithm.

The trajectory determined at the first level of the search, embodied by a string of xy coordinates, is then provided to the next level for refinement.

5. The Second Level: High Resolution Planning (Maneuver Planning)

The second level of refines the results of the first search by utilizing a similar search procedure but it operates on a reduced space in the vicinity of the first (approximate) plan and the graph generation process takes into account the body configuration, kinematics and dynamics of the vehicle in greater detail.

The vehicle model that is used at this second level includes the kinematic constraints of the vehicle as well as dynamical constraints on acceleration. Thus, the model of the second level utilizes the assumption that the steering angle and velocity of the model remain constant in the interval of modeling to predict the new coordinates of the vehicle. The kinematic model of Figures 1 and 2 is thus utilized in its final form. Also, a maximal rate of change of steering angle and velocity is used to describe the dynamics of steering and acceleration. A neighboring point is deemed to be unreachable from another if the transition requires a change of steering larger than allowed for a single transition.

In order to utilize the information provided by the rough plan of the first level, a neighborhood of this trajectory is constructed. The neighborhood is a closed subset of the workspace and is constructed as shown in Figure 3.

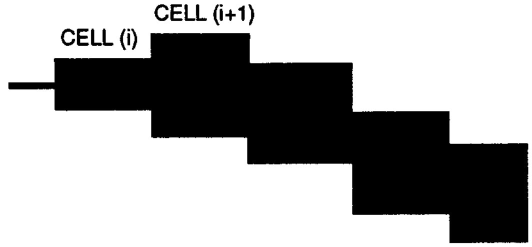

Figure 3. Construction of the vicinity of a rough trajectory.

The trajectory, which is described by a string of points, is enclosed by sequence of cells whose width is a function of the characteristics of the trajectory itself. As can be seen in the figure, straight segments are enclosed in narrow cells such as cell (i) whereas a change in direction establishes the need for greater width, as in cell (i+1). The area of the cells must be increased if complex maneuvers are expected (such as turns or direction reversal) because new constraints will be included in the new level of description and the abrupt changes which can be made with the slack constraints of the rough description may no longer be possible. The selective refinement of the scope of search is thus a heuristic for guaranteeing consistency in this paradigm. One further heuristic is applied at this stage; the cell size is increased if the baseline cell intersects an obstacle, because avoiding the obstacle may involve departing significantly from the nominal trajectory.

The number of dimensions in the search space is increased at the second level, where velocity becomes an additional axis. Thus, each cell from Figure 1 is decomposed into enough subcells to achieve the assigned average spacing of this second level and each randomly generated point has associated with it an x-position, a y-position, and a velocity.

A method of successor validation is utilized to determine children for each parent in order to preserve the fidelity of the representation of this level. This process begins with an evaluation of the steering angle 'α' which is required to attain each candidate successor point from each putative parent node. Inverse kinematic routines are used to determine this information as well as the corresponding center and radius of rotation. Constraints are applied in the following manner:

i] If the required steering angle exceeds the maximum steering angle, the successor is invalid.

ii] If the required *change* in steering angle exceeds the maximum allowable value, the successor is invalid.

iii] The time of transition is constrained by the maximum velocity allowed for the required radius of rotation. This value is determined from the maximal centripetal force which can be tolerated without slippage.

The edge cost is determined by the distance to be moved along the circumference of the circle and by the random velocity which is the third coordinate of the parent node. It has been determined experimentally that in order for the inverse kinematics routine to be numerically stable, the density of points in the XY-plane should be such that, on average, at least three points are located in a unit equivalent to the cross-sectional area of the vehicle.

One additional correction is to establish that the real volume of the vehicle, which is still represented by its center, does not intersect obstacles. For this computation, an area corresponding to the dimension of the vehicle is marked off around each valid node and tested for intersection with forbidden regions such as obstacles. In this manner, the graph representation of the second level can be modified to accommodate a realistic amount of detail in its model of the workspace.

Once more, upon completion of the graph representation of the workspace, the search algorithm is invoked to determine a path between the start and the goal positions in the graph. The resulting solution includes not only position trajectories, but velocity information and the steering angles required to follow the trajectory.

6. Results of Planning the Path of the Vehicle

Figures 4 and 5 depict the process of planning via snapshots of the screens presented to the user during planning. The process of search is shown in the left part of each Figure. Upper part shows the search in full space at low resolution. The lower part shows search in a reduced search space but at higher resolution. The final trajectory of the of the vehicle is shown in the right part of each figure for a workspace including a garage, wall, and gate.

The order of synthesis of this result can be seen beginning with Figure 4 which is a depiction of the search tree at the low level of resolution, overlaid on the description of the workspace. The kinematics of the vehicle are clearly absent from this consideration as can be seen by the result of search at the first level in the upper right part of the Figure 4 (the thin line trajectory) but are evident in the bold line trajectory in the same Figure which is the result of search at the next level: one can see the maneuvering of the vehicle. The search at this high-resolution level is depicted in the lower parts of Figures 4 and 5 where the reduced search tree of that level is shown.

This sequence of figures demonstrates that it is possible to synthesize complex maneuvers such as reversing and K-turns without using expert rule-base generated by a human being. Comparatively complex maneuvering is performed just by constructing a hierarchical representation of the system and searching for successive approximations to construct an ε-optimal solution of the problem.

7. Conclusions

The results shown in this paper demonstrate that multiresolutional search in a tesselated space can be successfully utilized for planning and control of an autonomous robot. The potential of this approach in the reduction of the complexity inherent in search techniques has been illustrated by a practical example. Maneuvering of the vehicle with no human generated rules have never been obtained before. It demonstrates that multiresolutional search can be a powerful tool for a truly autonomous robot planning and control.

References

[1] J. S. Albus, C.R. Maclean, A.J. Barbera, M.L. Fitzgerald, "Hierarchical Control for Robots and Teleoperators," *Proc. of the Workshop on Intelligent Control,* Troy, N.Y., 1985.

[2] A. Meystel *et al.*, "State Space Search for an Optimal Trajectory," Proc. of the *IEEE Int'l. Symposium on Intelligent Cont.*, vol. 2, Philadelphia PA, 1990.

[3] Y. Maximov and A. Meystel, "Optimum Design of Multiresolutional Hierarchical Control Systems," Proc. of the *IEEE Int'l. Symposium on Intelligent Cont.*, Glasgow, U.K., 1992.

[4] J. Pearl, "Heuristics," Addison-Wesley, Reading MA, 1984.

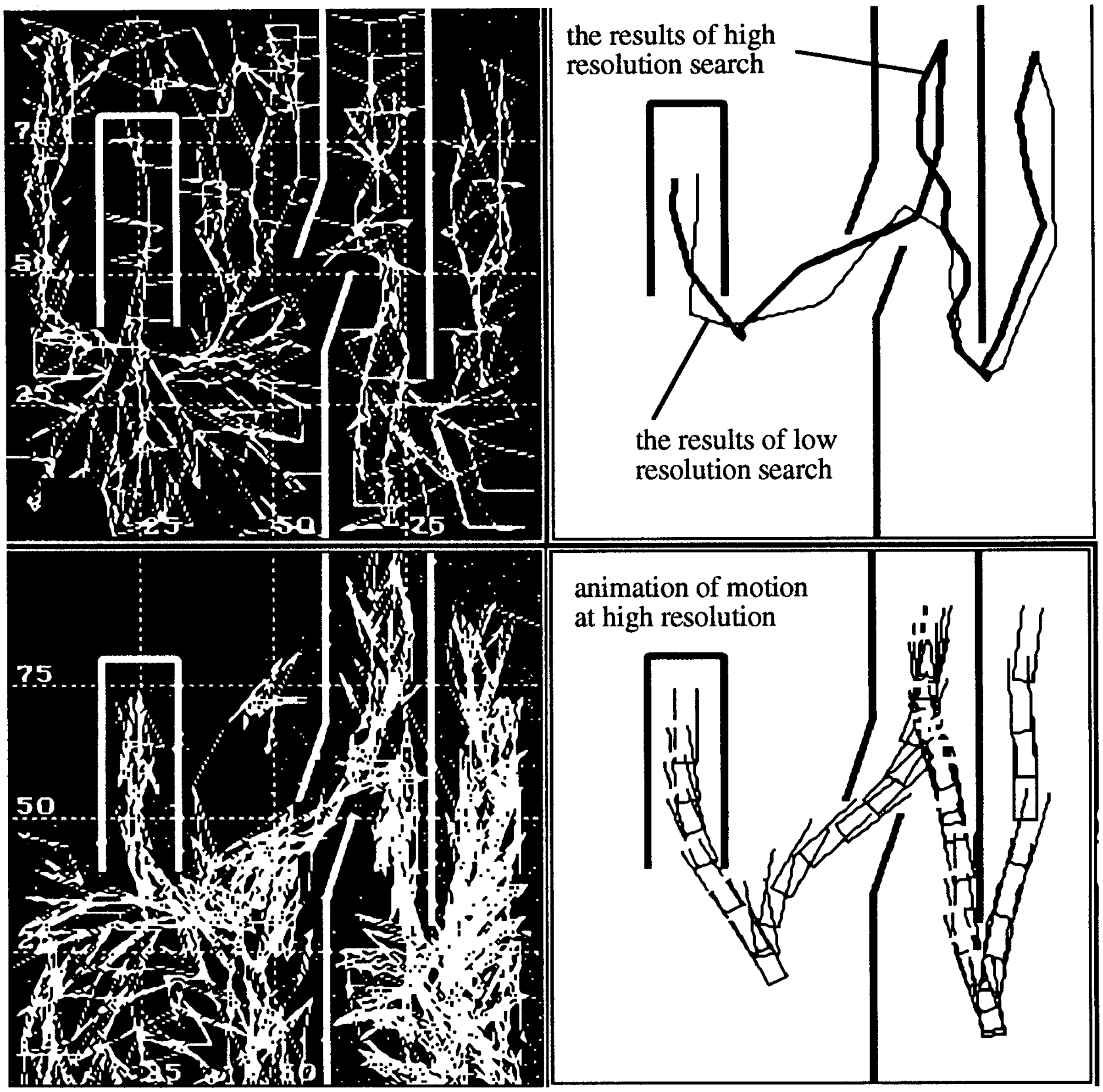

Figure 4. Experiment in a configuration 1 (a-search in the whole space at low resolution, b-search in the reduced space at high resolution)

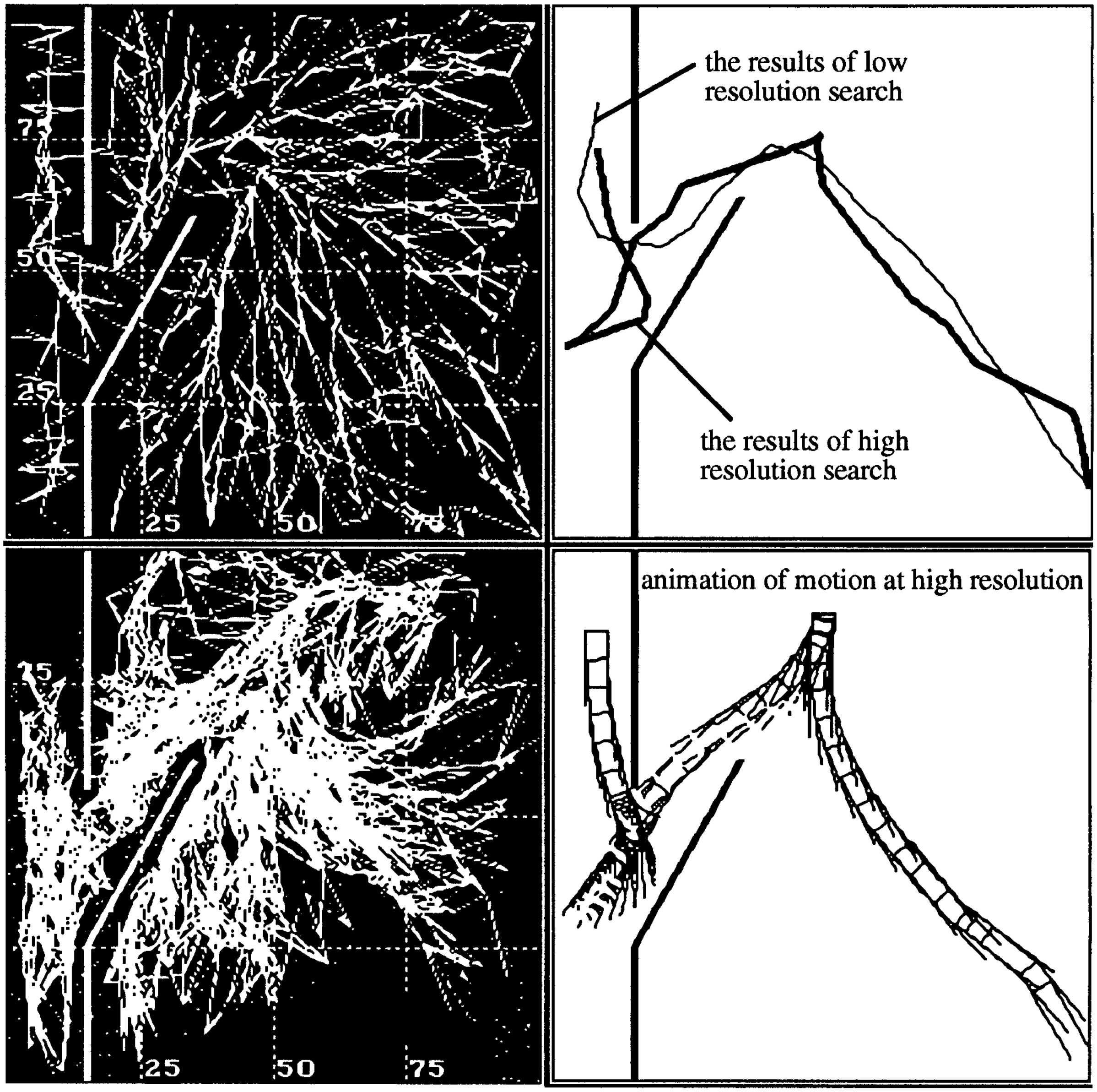

Figure 5. Experiment in a configuration 2 (a-search in the whole space at low resolution, b-search in the reduced space at high resolution)