MASSACHUSETTS INSTITUTE OF TECHNOLOGY
ARTIFICIAL INTELLIGENCE LABORATORY

A. I. Memo No. 736 **June, 1983**

Hypothesizing Channels Through Free-Space In Solving the Findpath Problem

Bruce R. Donald

Abstract. Given a polyhedral environment, a technique is presented for hypothesizing a channel volume through the free space containing a class of successful collision-free paths. A set of geometric constructions between obstacle faces is proposed, and we define a mapping from a field of view analysis to a direct local construction of free space. The algorithm has the control structure of a search which propagates construction of a connected channel towards a goal along a frontier of exterior free faces. Thus a channel volume starts out by surrounding the moving object in the initial configuration and "grows" towards the goal. Finally, we show techniques for analyzing the channel decomposition of free space and suggesting a path.

Acknowledgements. This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the System Development Foundation, in part by the Office of Naval Research under Office of Naval Research contract N00014–81–K–0494, and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014–80–C–0505 and N00014–82–K–0334.

1. Introduction

Channels are an encoding of free-space corresponding to the classes of paths within an environment. An implementation exploiting this global model of the connectivity of free-space has been able to solve 2-dimensional find-path problems in several minutes which formerly took many hours. Our algorithm is essentially a problem-solving strategy using a homeomorphic reduction of the search space.

Given a polyhedral environment, a technique is presented for hypothesizing a channel volume through the free space containing a class of successful collision-free paths. A set of geometric constructions between obstacle faces is proposed, and we define a mapping from a field of view analysis to a direct local construction of free space. The algorithm has the control structure of a search which propagates construction of a connected channel towards a goal along a frontier of exterior free faces. Thus a channel volume starts out by surrounding the moving object in the initial configuration and "grows" towards the goal. Finally, we show techniques for analyzing the channel decomposition of free space and suggesting a path.

This paper addresses issues in the *find-path* or *piano mover's* problem in robotics and spatial planning: the problem involves finding a path for a solid object in an environment containing obstacles. In robotics we are typically interested in motion planning for a mobile robot or manipluator. In Computer-Aided Design (CAD), the problem of automated structural design for n structural members is also an instance of the most general form of the mover's problem. A survey of robotics issues in robot motion planning can be found in Brady, *et al.* [3]. For related work on the mover's problem, see Brooks, [4], Lozano-Pérez [13, 14], Lozano-Pérez and Wesley, [15], Brooks and Lozano-Pérez [5], Schwartz and Sharir [23], Reif [21], Moravec, [16a], and Udupa, [26]. Some issues in automated structural design are addressed in Donald, [8]. For a review of geometric modeling techniques, see Baer, Eastman, *et al.* [1] and Requicha, [22].

1.1. Motivation

The primary motivation for this paper lies in the difficulty of the Find-Path, or "Piano Mover's" problem. In its most general form, with arbitrary degrees of freedom, the problem has been shown to be $\mathcal{P}$-*Space* hard.[1] (Reif [21]) With fixed degrees of freedom the problem is tractable but proposed algorithms have a high polynomial time complexity (Schwartz and Sharir [23], Reif [21]) and an implemented general path-finder for the 2-D mover's problem with rotations is quite slow (Brooks and Lozano-Pérez [5]).

Our observation has been that in general, local algorithms can get lost examining irrelevant local constraints. In particular, without adequate global knowledge of the connectivity of a workspace and the classes of paths it contains, such methods may choose impossible or ill-advised candidate paths. Thus channel hypothesis and path suggestion can serve as guidance for a more detailed method: we believe that the connectivity of configuration space may be inferred from the connectivity of real space.

The channel algorithm constructs a cell decomposition of free-space, which is then analyzed to determine the structure of the workspace and classes of paths it contains. We attempted to devise a method which formalizes previous approaches and generalizes to 3-dimensional workspaces. This paper represents a progress report on this work and a 2-dimensional implementation which illustrates many of the interesting general issues that arise in 3 dimensions. We will present techniques for constructing channel volumes and suggesting paths within them. We will also show in what way the find-path problem is "easier" in the transformed domain. An implementation of the algorithm for the two-dimensional mover's problem is described, and the results are discussed.

2. An Overview of the Algorithm

2.1. Criteria and Representations of Channels

Our idea is to transform the find-path problem from the domain of a multiply-connected free-space to the find-path-*containment* problem within a

[1] This is of more than theoretical interest. The CAD problem (above) for structural patterns or transformations requiring the movement or placement of n structural members is exactly this case.

simply-connected channel volume. Thus at one level the channel construction is a technique for characterizing the connectivity of the free-space; at another it is a geometric model for identifying classes of successful paths.

Let us begin by defining the criteria for channel volumes. A channel volume through an n-dimensional workspace embedded in $\Re^n$ should have the following characteristics:

(a) The channel volume should be simply-connected (whereas the free-space of the initial workspace is typically not).

(b) The channel volume should contain the $\Re^n$ projections of a class of successful paths. This formalizes the intuitive notion that a channel should "contain" a class of successful paths. Later in the paper we will formalize this criterion using aspects of homotopy theory.

(c) The constraints on motion within the channel should be *simpler*. Constraints on motion arise (in two dimensions) from vertices and edges on obstacles. However, it can be shown (Brooks and Lozano-Pérez [5], Lozano-Pérez [14]) that the constraints arising from concave vertices are subsumed by the neighboring edge constraints. Thus let us define the constraint complexity of a workspace as the total number of edges and convex vertices. Convex vertices generate the most complex constraints on motion. One competence measure for the channel approach to the mover's problem is reduction in convex vertex constraints in the transformed workspace.

Intuitively, these criteria make the transformed problem easier by providing an tightly-constrained "idea" of where to search within a complex workspace.

While we can take measures to ensure (a) and (b), it is possibly to construct cases where (c) will not hold. However, it should be clear that in complex workspaces simply-connected channel volumes can have far fewer edges and vertices than the initial environment.

3. The Channel Algorithm

Before we sketch out the algorithm, we need to have an intuitive idea of the channel *constructor*. Given two faces A and B on two obstacles, we wish to

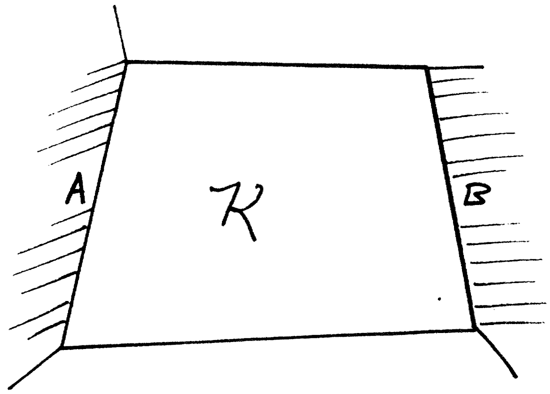

Figure 1. **The channel K constructed as the convex hull between faces A and B.**

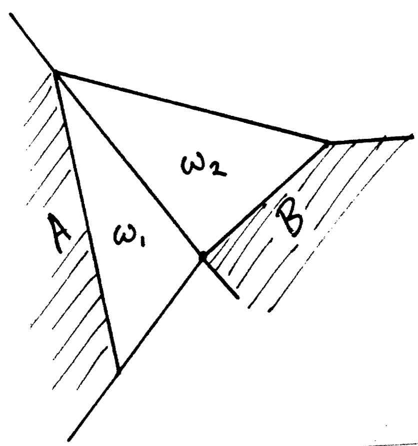

Figure 2. **A channel composed of the union of two wedges between A and B. $K = w_1 \bigcup w_2$.**

---

construct the region between them. We can think of this region as a "passageway" in free-space. First, a field of view analysis is performed to ensure that A has a clear view of B (or to determine what portion of B A can "see", and thus construct to). Next we perform a direct, local construction of free-space between the faces. The local constructions are used in a search expansion that propagates a cell-decomposition[2] towards the goal.

The direct, local construction of free-space is based on convex hull techniques and is performed by the channel constructor. Thus in two dimensions the region

[2]A *cell-decomposition* divides the space into a set of non-overlapping cells. We do not use cells of uniform size or shape, but instead employ convex regions constructed between faces.

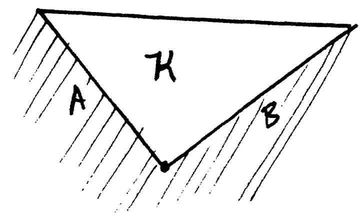

Figure 3. **A channel where A and B share a vertex.**

between A and B would typically be a convex quadrilateral, constructed by taking the convex hull of the faces.[3] We call this region a *channel* between A and B; our constructions will result in a cell-decomposition of part of free space, where each cell is a channel between two faces. The final simply-connected channel volume will be a contiguous path of these constructed cells.

Constructing such a channel introduces up to two *free faces* (see Forbus, [11]) into the environment,[4] and removes A and B from the environment as candidates for construction. Obstacle faces bound obstacle polyhedra, and free faces may have vertices on obstacles but do not bound obstacles.

The free faces, which are constructed as channel boundaries, may be *interior* or *exterior* to the entire channel decomposition. Thus an interior free face bounds two channel regions and an exterior free face bounds a channel region and "unknown", unexplored free-space.

During the construction, the outer boundary of the cell decomposition will contain both obstacle faces and exterior free faces. In a bounded, connected workspace, it is possible to construct a complete cell decomposition by searching until there are no more exterior free faces. In this case the union of the cells is

[3]This technique does not handle several important and common singular cases. We present a complete definition later.

[4]In the 3-dimensional case, each free face is triangular, and the number of free faces is the number of unshared edges on A and B.

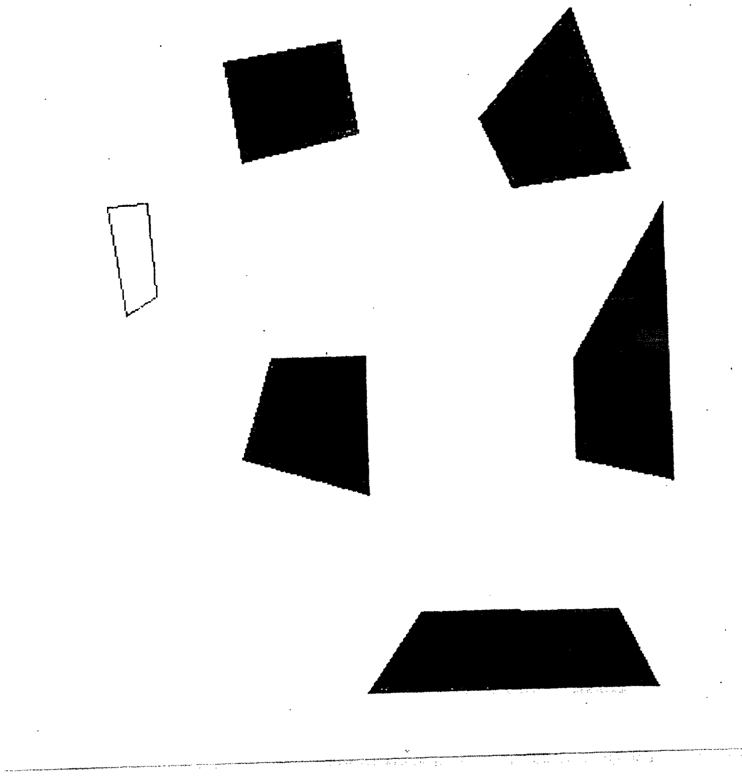

Figure 4. **A workspace before channel construction.**

equivalent to the initial workspace. Such a multiply-connected decomposition can always be searched for a singly connected cell path.

We are now ready to describe the algorithm: The *Frontier* of exterior free faces denotes the exterior free faces of the aggregate channel. The *Face Environment* contains all faces that bound only one region. (In our taxonomy, there are only two kinds of regions: *free* regions, and *obstacle* regions). Thus the Frontier contains candidates for direct, local constructions of free-space using the channel constructor, and the Face Environment is used as the input to the field of view computation. The algorithm has the structure of a search which propagates construction of a connected channel towards a goal along a frontier of exterior free faces. Thus a channel volume starts out by surrounding the moving object in the initial

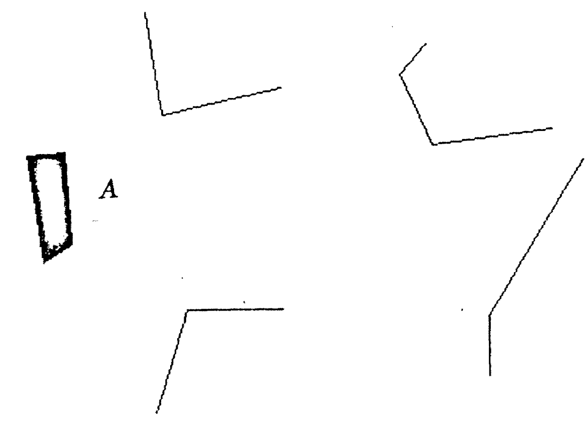

Figure 5. **The visible surfaces in the workspace from construction edge *A*.**

configuration[5] and "grows" towards the goal.

For each expansion in the search we examine the frontier of the channel decomposition and choose the "best" free face to construct from. Next we determine what faces or portions of faces in the face environment it can construct to, and choose the best region. We construct a channel, add it to the channel decomposition, and update the frontier and face environment appropriately. The construction halts when the moving object in the goal configuration is contained in the channel decomposition.

Our intuitive development of the channel constructor used examples of channel construction between obstacle faces. Note that although this is possible, in the

[5] The *configuration* of a rigid polyhedral object (see Lozano-Pérez, [13]) is a set of parameters representing the combined translation and orientation of the object. Thus for example the configuration of a polygonal object with two translational and one rotational degrees of freedom is typically represented by the parameters (x, y, θ). A path for the polygon is a sequence of such configurations.

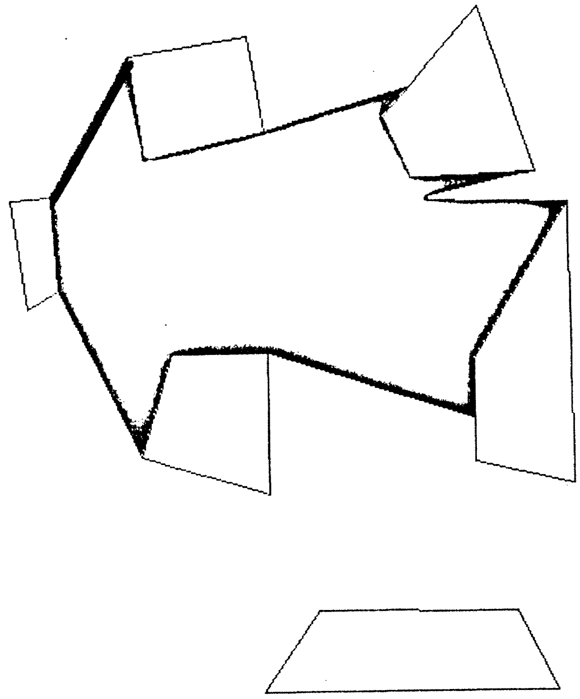

Figure 6. **The union of all candidate channel regions that can be constructed from A.**

search formulation above we actually construct channels between a *free* face on the Frontier and a (free *or* obstacle) face in the Face Environment.

3.1. The Algorithm in more detail

Phase I: Channel Construction

(1) We construct a polyhedron P around the "piano" (or moving object) in the start configuration. The channel decomposition (set of all channels) is initialized to be this polyhedron, and the frontier of free faces is initialized to be its faces. The Face Environment is initialized to contain all obstacle faces in the initial workspace plus the faces of P.

(2) Select the best face A on the frontier. (H1)[6]

[6] *Heuristic selection criteria* are required in the algorithm, and are denoted by (H1), (H2), etc. The implementation of these criteria is discussed later.

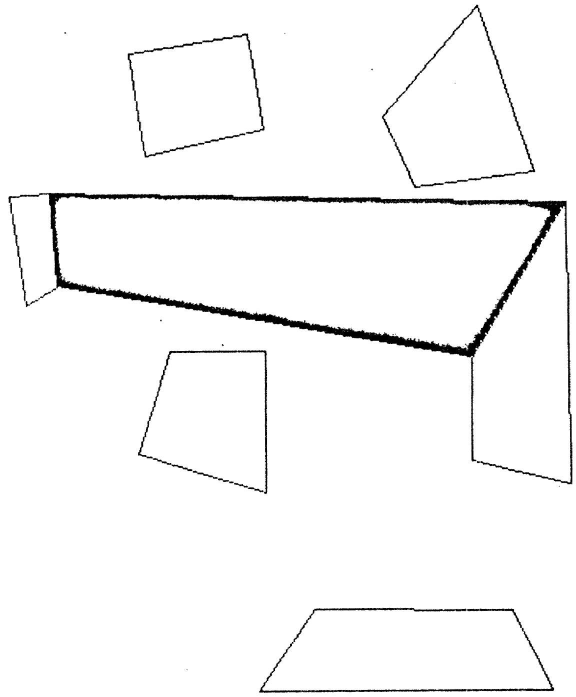

Figure 7. **The best channel is chosen.**

(3) Perform a field of view analysis from A to determine the set of visible regions in the face environment. These are the candidate construction regions, and correspond intuitively to all the faces or portions of faces that "A has a clear view of." Out of these select B, the best construction region. (**H2**)

(4) Construct a channel $\mathcal{K}$ between A and B.

(5) Update the search frontier and face environment. This is done as follows: The boundary of $\mathcal{K}$ contains A, B, and a set of free faces F (in the two dimensional case, F contains one or two faces). The frontier may only contain *exterior* free faces of the channel decomposition, and the free faces F might not be exterior to the channel if they are shared by a previously constructed region. However, it is easy to distinguish the interior free faces by examining their coboundary. The *coboundary* of an n-cell κ (Giblin, [12a])) consists of the set of $(n+1)$-cells it bounds, and is denoted $Co\beta(\kappa)$. We can keep track of the coboundary of each face by recording what solids it is used to construct. The *cardinality* of the coboundary of interior

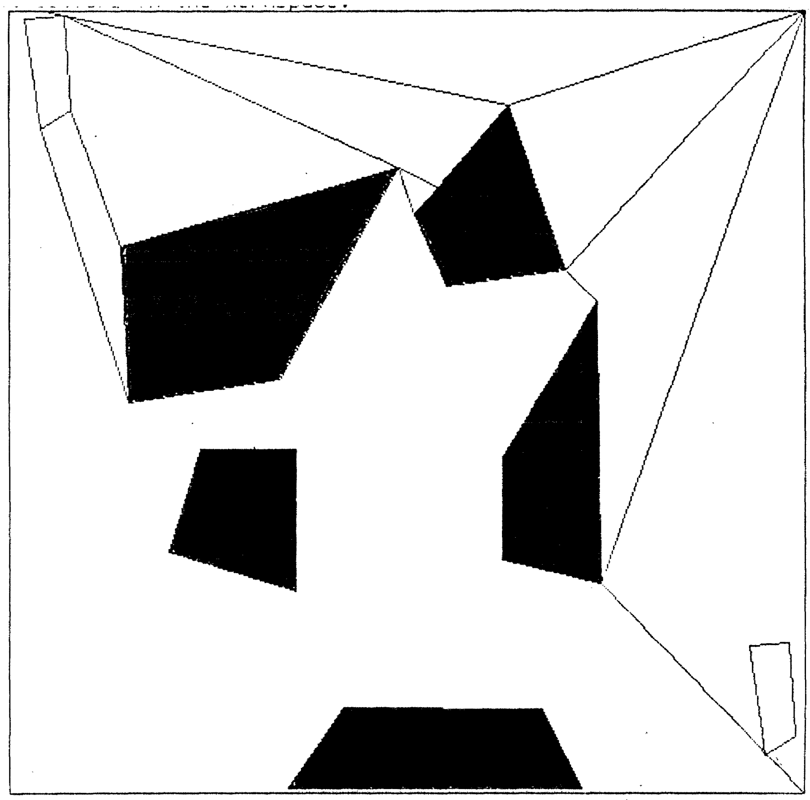

Figure 8. **A Workspace showing a channel decomposition containing the moving object in start and goal configurations**

free faces is 2 (since they bound two constructed regions) whereas the cardinality of an exterior free face's coboundary is 1. Thus our update is very simple:

Let *Env* denote the Face Environment and *Front* the Frontier of exterior free faces.

Delete A from the *Front* and *Env*.

Delete B from *Env*. If $B \in Front$, delete B from *Front*.

For each face f in F (recall that F is the set of free faces of $\mathcal{K}$): if f is exterior (i.e., if $|Co\beta(f)| = 1$) then add f to *Front* and *Env*. Otherwise, f was an exterior free face prior to the construction of $\mathcal{K}$, and is now interior to the channel decomposition and no longer on the frontier: delete f from *Front* and *Env*.

(6) If the channel $\mathcal{K}$ or the union of all channels contains the moving object in the goal configuration, stop. Otherwise repeat steps (2–6).

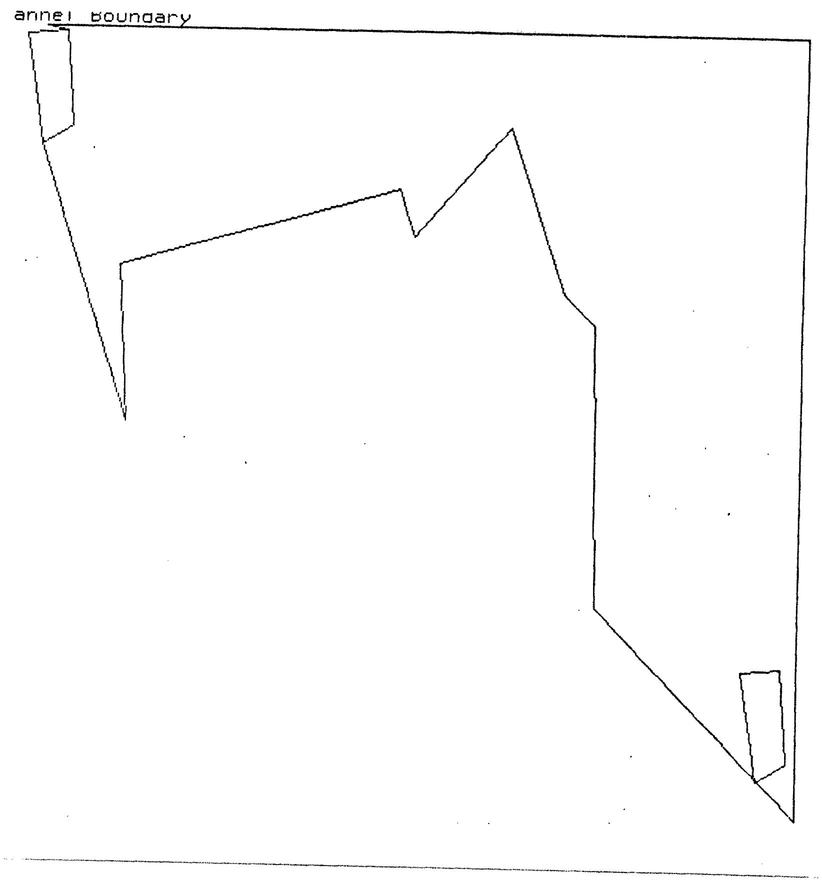

Figure 9. **The outer union of the channel volume from figure 8. Note the reduction in constraint complexity.**

Phase II: Path Analysis

(1) Phase I has constructed a decomposition of the workspace into a connected set of non-overlapping cells. This decomposition may be multiply-connected. Thus we search the cell decomposition for the best *cell-path.* **(H3)**

(2) The cell path is now our simply-connected channel volume. Identify and label the constraints to motion imposed by the channel.

(3) Suggest a configuration path through the cells. **(H4)**

(4) The configuration path consists of a sequence of configuration points within the channel. Verify each configuration on the path to see whether it is legal. If the configuration is blocked, attempt to rectify it through small corrections in orientation. Interpolate between verified configurations to a given resolution.

The effect of step (4) is a partitioning of the resulting find-path problem within the hypothesized channel into simpler subproblems along a visibility graph.

4. The Channel Constructor

In this section we formalize the concept of the channel constructor, which is used to build a local channel region between two faces. Note that in the algorithm above, at least one of these faces is always an exterior free face on the channel frontier.

To construct the region between two faces, we will use the *convex hull* (Grünbaum [12b]) of the vertices of the faces. This works well when the region between two faces is convex: but for arbitrarily positioned faces this is not always the case, and the convex hull can intersect the interior of the obstacles. (See Fig. 2). Thus we adopt a recursive definition in which we try to construct a convex hull between cells of maximum dimensionality. For example, if the region between two n-faces[7] is non-convex, then we can model it as the union of two *wedges* between n- and (n-1)-cells.

We define the following geometric constructions in order to model the region between faces embedded in n-space, where n is 2 or 3. The *pyramid constructor* constructs the region between a face and a point. A *wedge* is identical to a pyramid in two dimensions, but in three dimensions is used to model the region between a 2-face and an edge (1-face). We first give a generalized definition for 3 dimensions, and then interpret it for the both the two and three dimensional case.

4.1. Conventions

Let us adopt the following notational conventions: $i(X)$ denotes the interior of a set X and $\beta(X)$ its boundary. $conv(X)$ is the *convex hull* of a set X. $vert(X)$ is the set of vertices of a polyhedron X.

4.2. The Definitions for 3 Dimensions

Def: The *pyramid constructor* $\mathcal{P}(f,v) = conv(vert(f) \bigcup \{v\})$ where f is an (n-1)-face and v is a vertex in general position embedded in n-space.

[7] n is the dimensionality of the workspace, and a point is a 0-cell, a line a 1-cell, etc.

Now, let A, B be convex (n-1)-faces of the n-polyhedra P_a, P_b, embedded in n-space. Assume that neither face lies entirely behind the plane of the other. (See section 5). Let e_a, e_b be (n-2)-faces of P_a, P_b such that $e_a \in \beta(A)$ and $e_b \in \beta(B)$.

Def: The *wedge constructor* $\mathcal{W}(A, e_b)$, where $A \bigcap e_b = \emptyset$, is defined as follows:

Let $w = conv(vert(A) \bigcup vert(e_b))$. If $w \bigcap i(P_b) = \emptyset$ then $\mathcal{W} = w$. Otherwise $\mathcal{W} = \mathcal{P}(A, v_b)$ where v_b is the closest vertex in $vert(e_b)$ to A. This can only occur in 3-dimensional (or higher) space.

Def: The *channel constructor* $\mathcal{K}(A, B)$ is defined as follows:

Let $\kappa = conv(vert(A) \bigcup vert(B))$. If $\kappa \bigcap i(P_a) = \emptyset$ and $\kappa \bigcap i(P_b) = \emptyset$ then $\mathcal{K} = \kappa$. Otherwise $\mathcal{K} = \mathcal{W}(A, e_b) \bigcup \mathcal{W}(B, e_a)$ for some appropriately chosen e_a and e_b.

4.3. The Definitions for 2 Dimensions

In two dimensions, A and B are edges bounding polygons, and e_a and e_b are vertices. Hence:

In two dimensions, a pyramid is simply a triangle.

In two dimensions a wedge is exactly a pyramid. We use wedges to partition non-convex regions between faces.

4.4. Interpretation of the Channel Constructor

We interpret the channel constructor as follows: consider the two dimensional case first. If the region between A and B is convex, then we construct it directly. If it is non-convex, (i.e., if the convex hull intersects the interior of the regions A or B bounds), then we construct a region by building a wedge from A to a vertex ((n-2)-cell) on B and a wedge "back from" B to a vertex on A. In two dimensions the selection of e_b and e_a is trivial since each edge contains only two vertices. In three dimensions e_b may be chosen arbitrarily. However, we can heuristically choose a large edge on B to maximize the size of the construction (see appendix I). The construction of the first wedge makes the choice of e_a deterministic and the wedge union may be performed using combinatorial techniques requiring no geometric intersection. Wedge construction in three-dimensions is briefly addressed in an appendix. Examples of 3-dimensional channel constructions are shown in figures 28 and 29.

Since all of the points in the hull set are initially embedded in faces and edges, it is possible to construct all of these convex hulls in $O(n)$ time for n edges (Preparata and Hong, [20]). Of course in two dimensions, the construction time is constant because there are only 2 vertices per edge.

4.5. A Constant-Time Constructor for the 2-Dimensional Case

The channel constructor can be expressed as a very simple algorithm in two dimensions. We can construct the region between A and B directly, check it for convexity, and partition it into two triangles if necessary. The construction amounts to determining the free faces from two choice sets on the graph of vertex connections, which can be done by simply minimizing the sum of their lengths. In the case of non-convexity, the resulting quadrilateral can be partitioned into wedges by constructing an edge from the concave "notch" (Chazelle, [7]) to the opposite vertex.

Finally, to construct a channel region between two faces that share a vertex, we simply build a triangle after introducing one free edge.

5. Field of View Analysis

In this section we address the field of view analysis performed in the construction-propagating search. We perform the field of view computation as a *sight-line* analysis to determine what surfaces a frontier face can construct to.

The field of view analysis contains two components: a visible-surface computation to determine candidate construction regions for a frontier face, and a heuristic selection from candidates in the image-plane. The heuristic selection of a candidate region is based on geometric criteria.

A number of different mechanisms (for example, plane sweep algorithms; see Nievergelt and Preparata, [16b]) could have been used to determine constructible regions from a frontier face. The field of view techniques were adopted for the following reasons:

(1) They were extensible to three dimensions, where the problems are well understood and efficient algorithms abound. Plane sweep algorithms are difficult to extend to three dimensions, especially for general polyhedra. Three-dimensional retraction algorithms (See Ó'Dúnlaing and Yap, [18], Ó'Dúnlaing, Sharir and Yap [19]) have not yet come into existence, and present other problems which we discuss in section (10.1). In particular we should note that field of view algorithms are relatively insensitive to minor geometric variations (unlike Voronoi diagrams, for example).

(2) The field of view method allows an implementation of selection criteria to operate almost exclusively in a lower dimension (the image plane). This allows us to abstract out qualitative geometric characteristics with less computational overhead.

(3) A fundamental step in the channel method involves the partitioning of the resulting find-path problem within the hypothesized channel into simpler subproblems along a visibility graph. A field of view computation seems a natural means of enforcing the visibility graph constraints on the construction of a channel around a suggested path. In particular, the initial nodes along the visibility graph of suggested configurations can be located on the centroids of mutually visible interior channel faces.

(4) The field of view algorithm allows local constraints to be captured in the construction of a global decomposition of free-space. The channel method is a construction of a new constraint space; in this new workspace we wish to minimize the total number of constraints and introduce as few artificial constraints (exterior free faces) as possible. This means that constraints from local obstacle surfaces must be incorporated into the channel boundary. Our implementation of selection criterion (**H2**) allows local constraints to be captured through anti-fragmentation heuristics (See Section (6.1)).

(5) Finally, field of view analysis is a good technique for ensuring the local

convexity[8] of the decomposition. The sight-line criterion (that a surface must be visible from the frontier) in conjunction with the channel constructor enforce this constraint on the local constructions of free space.

Once more, we will begin with a general discussion of 3-dimensional field of view analysis, and return to describe the two-dimensional implementation.

5.1. Visible Surface Algorithms

Visible surface algorithms are ubiquitous in the computer graphics literature. (See, for example, Sutherland, Sproull, *et al.* [25], Foley and van Dam, [10]). We perform a visible surface computation with the vantage-point "on" or slightly inside a frontier face A of the aggregate channel to determine a "scene" or *view* of obstacle faces and exterior channel faces in the Face Environment. Given the "image" or representation of the field of view from A we then want to heuristically select the best construction region (which will be the projection of a face or portion of a face B), invert the projection to find the world coordinates of the region B, and construct the channel $\mathcal{K}(A, B)$.

It is crucial to realize that we select the construction region based on its *image* in the picture plane, and then invert the perspective transformation to compute the corresponding region in the workspace.

There are two special criteria for the visible surface computation: it must be *reconstructive* and *automorphic.*[9] A reconstructive algorithm computes coherent, (n-1)-dimensional regions (polyhedra) that are visible in an n-dimensional scene. Thus scan-line and "painters" algorithms are not reconstructive. The algorithms of Sechrest and Greenberg, [24] and Wittram, [27], for example, are reconstructive in that they compute a list of visible polygons for a 3-dimensional scene.

The perspective transformation, since it is a projection, is not naturally invertible. We need an invertible transformation in order to map back to the

[8]Many problems in geometric modeling are much easier for convex or locally convex objects. In particular, the find-path problem is easier within a locally convex volume.

[9]Automorphism in geometric modeling is a common information-preserving technique. For one of the more elegant introductions to algorithms involving automorphic transformations, see Kalay, [12d].

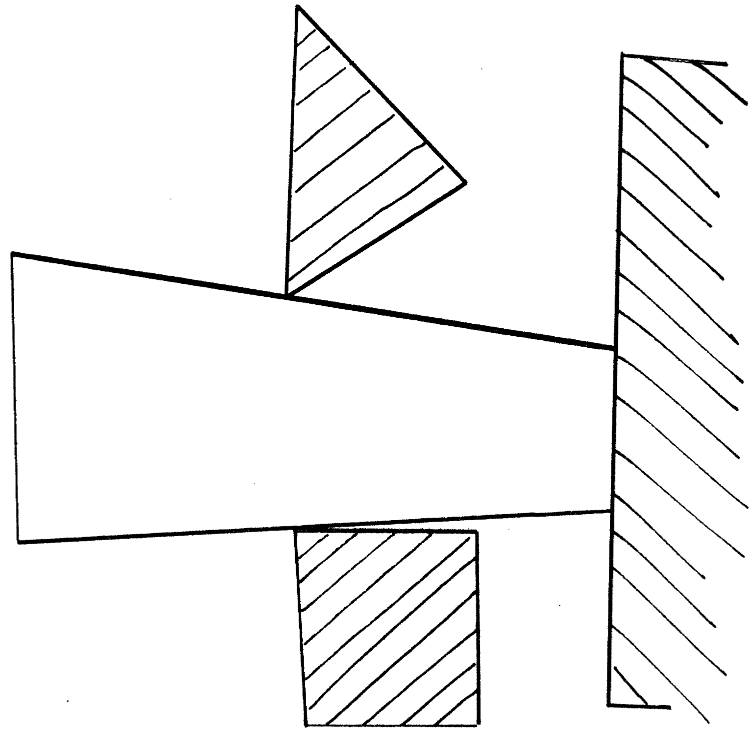

Figure 9a. **Partially obscured surfaces may be fragmented by construction to the visible portion.**

workspace from the field of view. Thus we make the reconstructive algorithm automorphic by maintaining in parallel the generator or interpolated position for each projected vertex or edge intersection.

For the two-dimensional case the visible surface algorithm is much less complex, and amounts to computing one "scan-line" of visible edges. However, reconstructive 3-dimensional visible-polygon techniques such as that of Sechrest and Greenberg, [24] are well understood.

5.2. Fragmentation

The surfaces visible in the field of view will be portions of surfaces in the workspace. Objects are partially visible when they are obscured by other objects, too large to fit into the field of view, or partially behind the picture plane. If we construct to one of these regions we fragment the containing region in the workspace. Fragmentation is not difficult to deal with in the 2-dimensional case since construction to a visible region can (1) consume all of an edge, (2) consume "one end" of an edge, leaving behind the other, unconstructed end, or (3) consume the "middle" of an edge, splitting the edge into two unconstructed segments separated by an obstacle face of the channel. We call cases (2) and (3) *fragmentation* of the workspace surfaces, and in fact our selection heuristics on visible regions in the field of view attempt to avoid fragmentation whenever possible.

In the 3-dimensional case fragmentation will be more of a problem. In particular we note that the construction regions A and B must always be convex; but even if all workspace faces are convex, the visible regions may be non-convex due to obscuration. Similarly, even if a visible construction region is convex, when constructed to it may well leave a "hole" or a non-convex "notch" in the containing region. In principle these issues can be dealt with through arbitrary triangulation of visible faces. A better solution will involve optimal or near-optimal convex decompositions such as those proposed by Chazelle, [7], and a set of heuristic preference criteria over a taxonomy of fragmentation-producing constructions. For example, three-dimensional anti-fragmentation heuristics would typically prefer convex constructions that partitioned containing regions convexly. A complete taxonomy is beyond the scope of this paper, but the geometric tools for convex decomposition exist and it can be shown that the anti-fragmentation preference criteria contain at most 16 equivalence classes.

5.3. Implementation

In the two-dimensional implementation, we perform a visible-surface analysis from each vertex of the Frontier edge, and then intersect the visibility constraints to determine the constructible field of view. A subsumption criterion which compares edge visibility constraints is also employed to detect when regions are visible from the vertices but not from the interior of the segment.

6. Searching Strategies

The control strategy for channel construction has the structure of a search. In this section we discuss characteristics of the search and of the heuristic selection criteria employed. These criteria, (H*i*) are implemented as evaluation functions which define a partial order on the candidates.

The search has several stages:

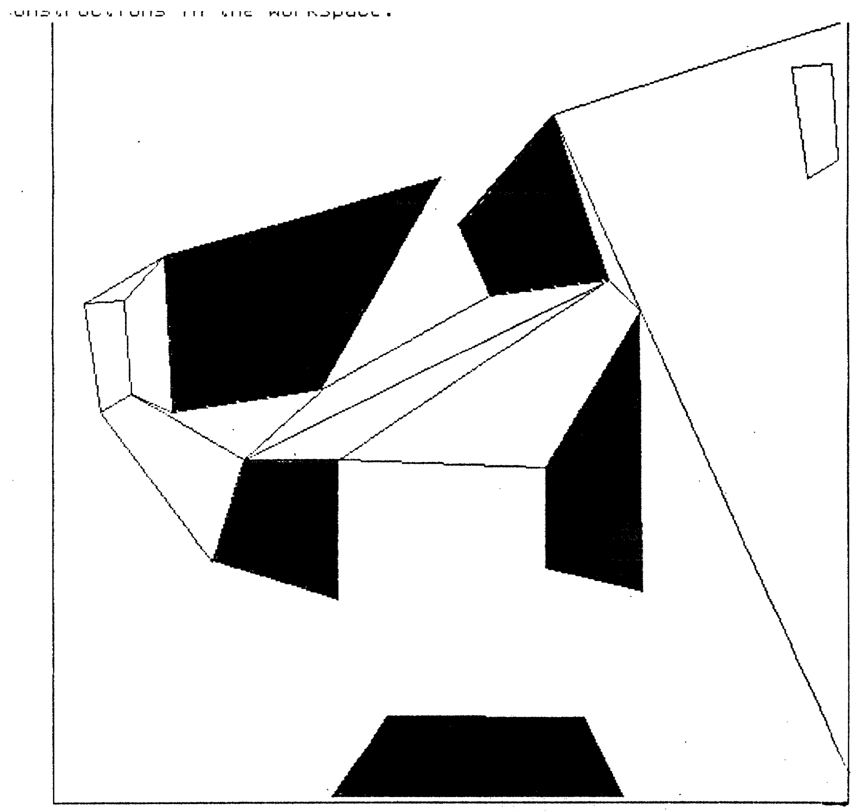

Figure 10. **Another channel decomposition for a different find-path problem.**

Construction propagates as a *best-first* search (Nilsson, [17]) along a frontier of exterior free faces, and each channel construction adds and deletes surfaces from the Frontier (*Front*) and Face Environment (*Env*). (H1) is used to select the best face A on the frontier for the next local channel expansion. When the visible surface computation is completed, heuristic (H2) is used as an evaluation function on the surfaces in the field of view to select the best "matching" construction region B. Finally, at the end of all construction, the cell decomposition is searched (again using a best-first techniques) for the a cell path. (This cell path is called the *channel path*). (H3) is used as an evaluation function for the cell path search. Note that it would also be possible to use an A^* search, (Nilsson, [17]) using these heuristic notions of optimality over the channel path.

6.1. Heuristic Selection Criteria

In general the selection criteria prefer "large channel constructions that make progress towards the goal." The heuristics structure the representation of free-

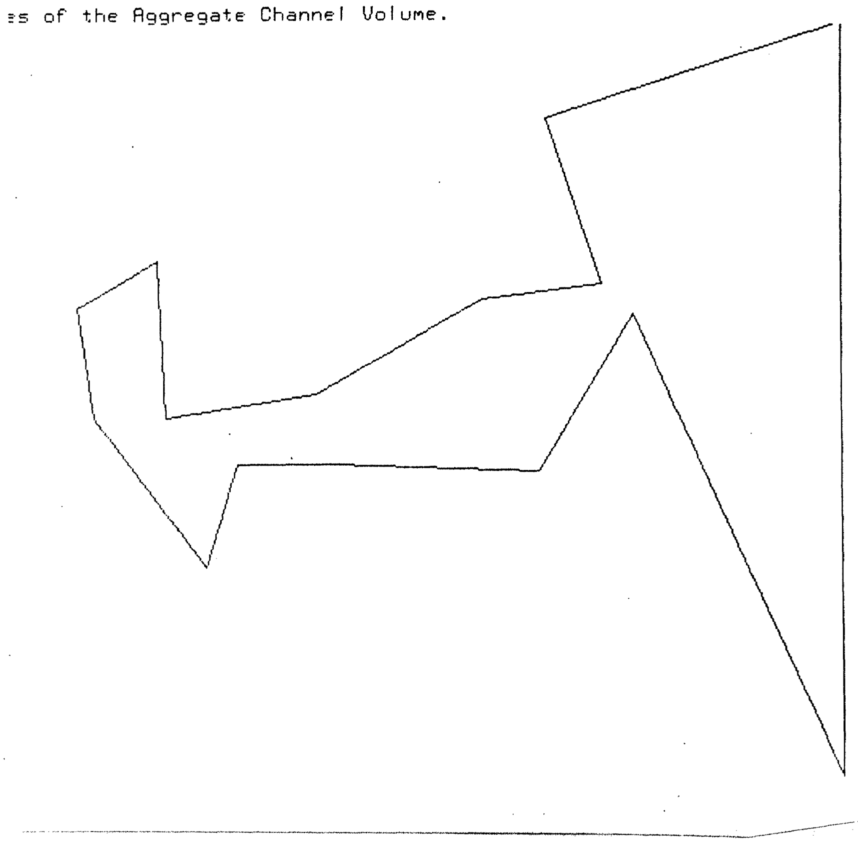

Figure 11. **The outer union of the decomposition of figure 10.**

space. For example, in choosing goal-facing construction surfaces, the heuristics can identify channels containing paths in the right direction.

(H1) is the evaluation function which selects the best construction face on *Front* using the criteria of *progress* and *face-size*. The exterior free faces are classified by goal-proximity[10] into neighborhoods. Within the closest neighborhood the largest face is selected. If no suitably large face exists in the neighborhood the next closest neighborhood is examined until a construction face is found. In addition, **(H1)** prefers frontier surfaces that face the goal. Thresholding is used in metric comparisons so that minor variations in size or goal-proximity are not over-weighted.

(H2) defines a partial order on visible face regions in the field of view. Since the visible surface computation is automorphic, **(H2)** has access to the workspace attributes of these (n-1)-dimensional polyhedra. The progress and size criteria of

[10] For distance to the goal we use Euclidean distance from the (x, y) projection of the goal configuration.

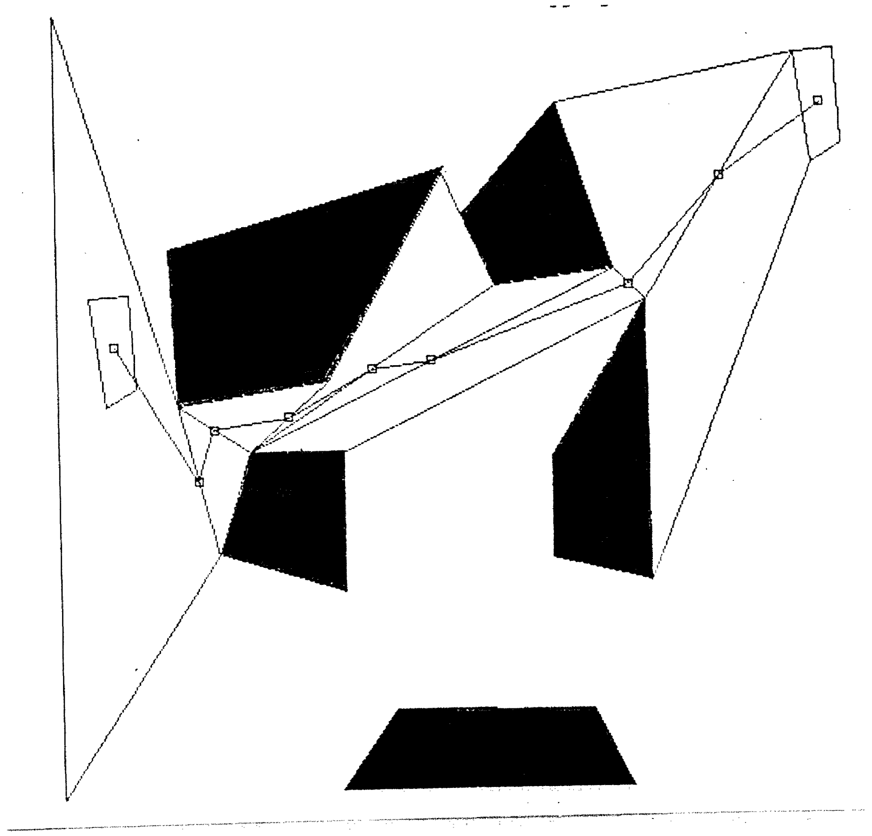

Figure 12. **A path suggested through the channel.**

(H1) are used; in addition (H2) attempts to minimize fragmentation. Recall that A is the selected face on *Front*. (H2) prefers faces which are not split by extensions of A (i.e., faces not intersecting the plane of A). (H2) also minimizes construction fragmentation, preferring construction to entire faces if possible. If an edge must be split, then (H2) favors fragmentation into one split edge over two.

The anti-fragmentation heuristics have the following effects: At each stage of construction, we expand the channel to the unfragmented face that makes the most progress towards the goal. Instead of constructing long, narrow channels making a lot of metric progress towards the goal, the search is attracted by local faces which are entirely visible. Thus there is a tendency to maximize the breadth of the channel towards the containing obstacle surfaces. The aggregate channel boundary tends to incorporate these local obstacle surfaces instead of building exterior free faces that "skim" the obstacles. This helps avoid artificially narrow channels. (H2) is designed to cause the channel constraints to be *inherited* from the original workspace (and

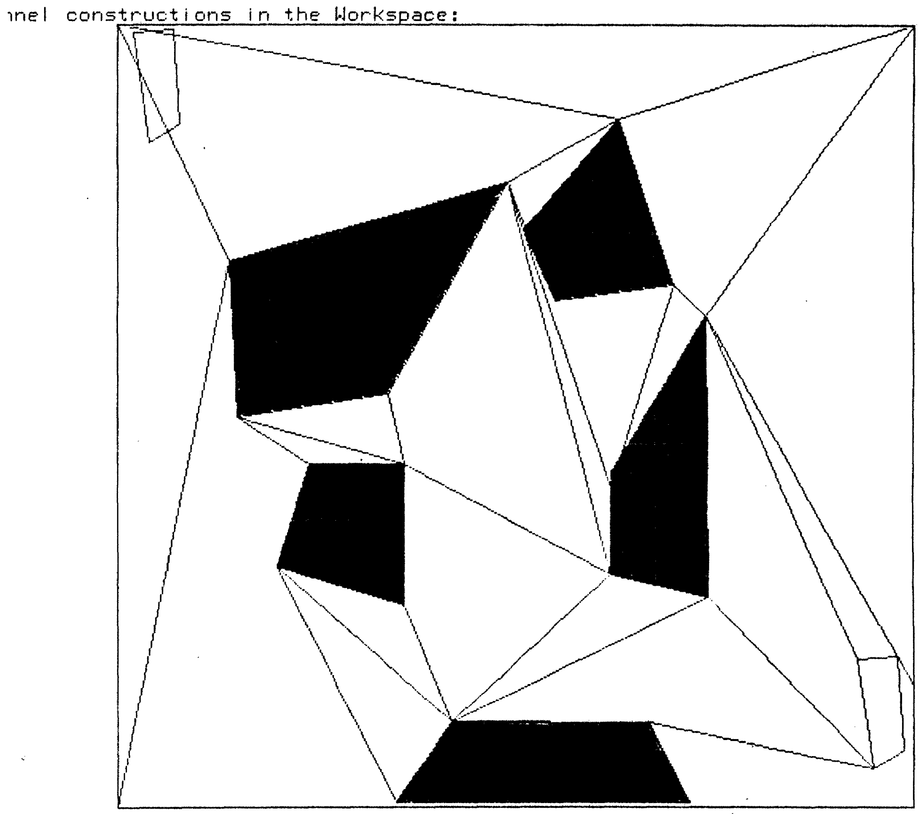

Figure 13. **This example shows a *multiply-connected* channel decomposition.**

not artificially introduced), and to maximize the locally available free space within a channel.

6.1.1. Backtracking and Thresholding

The search control maintains a simple library of cross-sections of the moving object, and uses these as a *threshold* for channel construction. Thus if the field of view analysis can only find extremely small channels visible from a frontier surface, then the search is aborted and a new frontier face is found. Thus not only does the search attempt to maximize the size of the channel interfaces, but heuristically detects when these interfaces are singular or too small.

6.1.2. Searching the Channel Decomposition

The Structure of the Channel Decomposition

The channel decomposition is a connected set of polyhedral channel regions. These channel regions adjoin along interior free faces (the *interfaces*). A *start* and

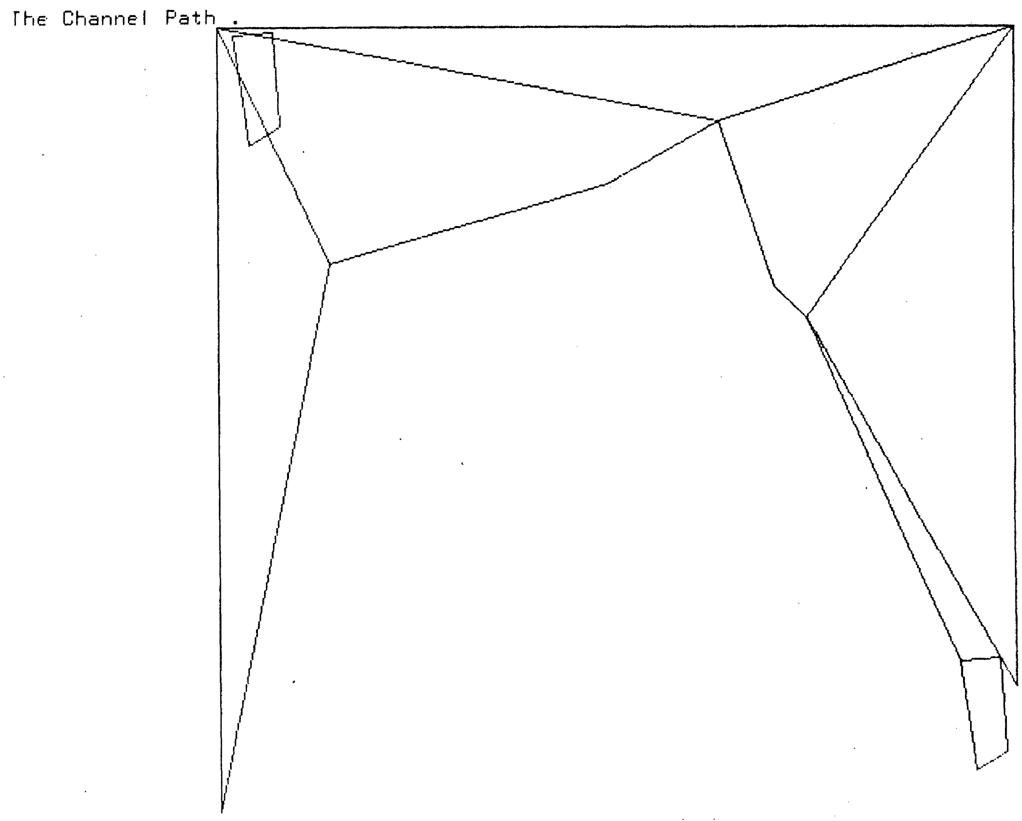

Figure 14. **The coverage is searched for a channel path.**

goal channel are designated as follows: the start channel is the bounding polyhedron P, built around the moving object in the initial configuration.[11] The *goal* channel is defined to be the union of all channels containing some part of the object in the goal configuration.

(H3) is the evaluation function for the final search through the cell decomposition for a simply-connected channel path. It guides the search by attempting to maximize the size of the interface, thus choosing for search expansion the channel with the largest connecting interior face.

7. Path Analysis

In the Path Analysis phase of the algorithm we analyze the simply-connected channel volume and identify and label all constraints to motion. Next, a path

[11] In our implementation, we construct P to be slightly larger than the object, thus constructing some manœuvring room near the initial configuration.

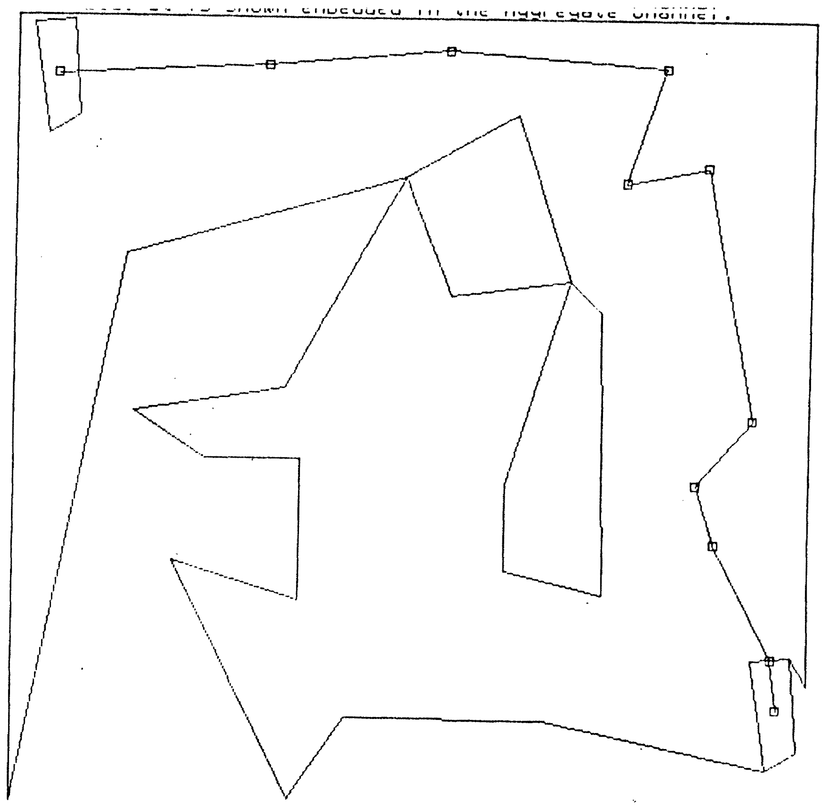

Figure 15. **The suggested route through the channel in relation to the outer union of *entire* decomposition. This case demonstrates the necessity of the post-construction search for a cell path.**

is suggested through the channel cells. We employ configuration space (*C-Space*) techniques (Lozano-Pérez [13, 14], Lozano-Pérez and Wesley, [15], Brooks and Lozano-Pérez [5]) to verify the suggested path and to rectify nodes on the path which cause collisions. Finally, the path is interpolated to a given fine-grain resolution.

7.1. Suggesting a Path through the Decomposition

To suggest a path through the simply-connected channel volume we place the reference point at the centroid of the moving object and attempt to move it through the centroids of shared interior faces. In two dimensions we adopt the following sub-path techniques:

Moving between two contiguous edges: For a triangular channel, suggest a sub-path through the midpoints of the edges. In a quadrilateral, choose a *via* point on the midpoint of the shared diagonal.

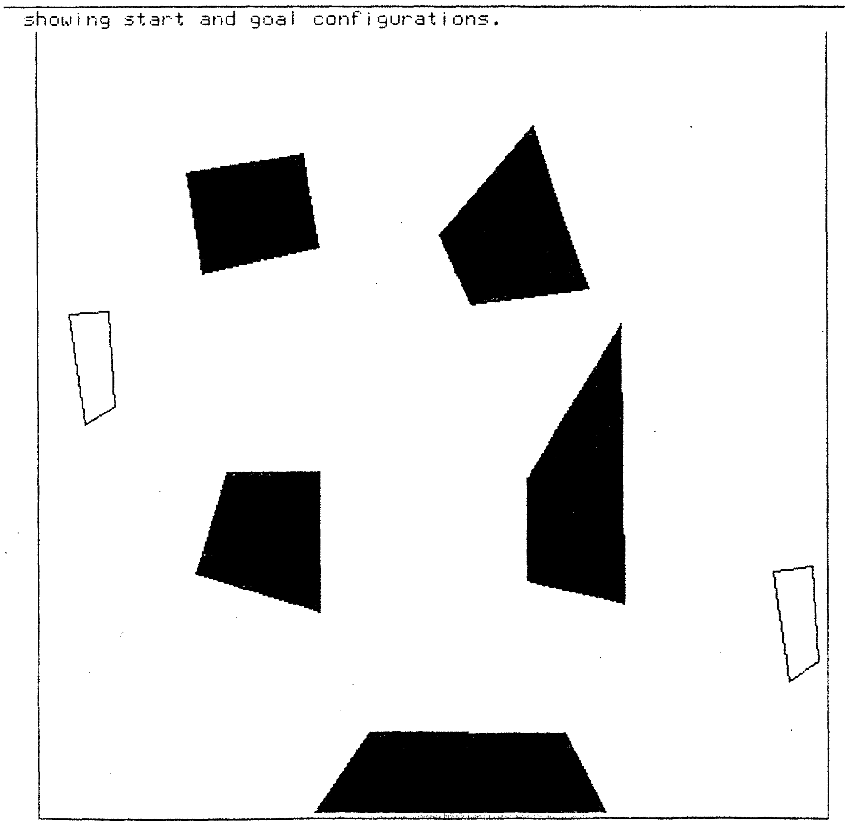

Figure 16. **Another find-path problem.**

Moving between two disjoint edges of a quadrilateral: Suggest a sub-path through the midpoints of the edges and the centroid of the quadrilateral.

This is a simplified subset of the sub-path heuristics ((H4)). The implementation also considers interfaces of two contiguous edges and certain singular cases.

7.2. Suggesting Rotations Along the Path

More complex path-suggesting techniques are possible. For example, it is possible to compare the interfaces to a library of cross-sections of the moving object, and select an orientation that will "fit." This is of course complicated by the changing orientations of the interfaces along the channel path. Such techniques have not been implemented, partially because one of our goals was to show that even very crude path suggestion is useful in solving the mover's problem. Our emphasis here is on the channel itself: Given the constructed channel volume containing a class of paths, it should not be hard to refine the path suggestion techniques.

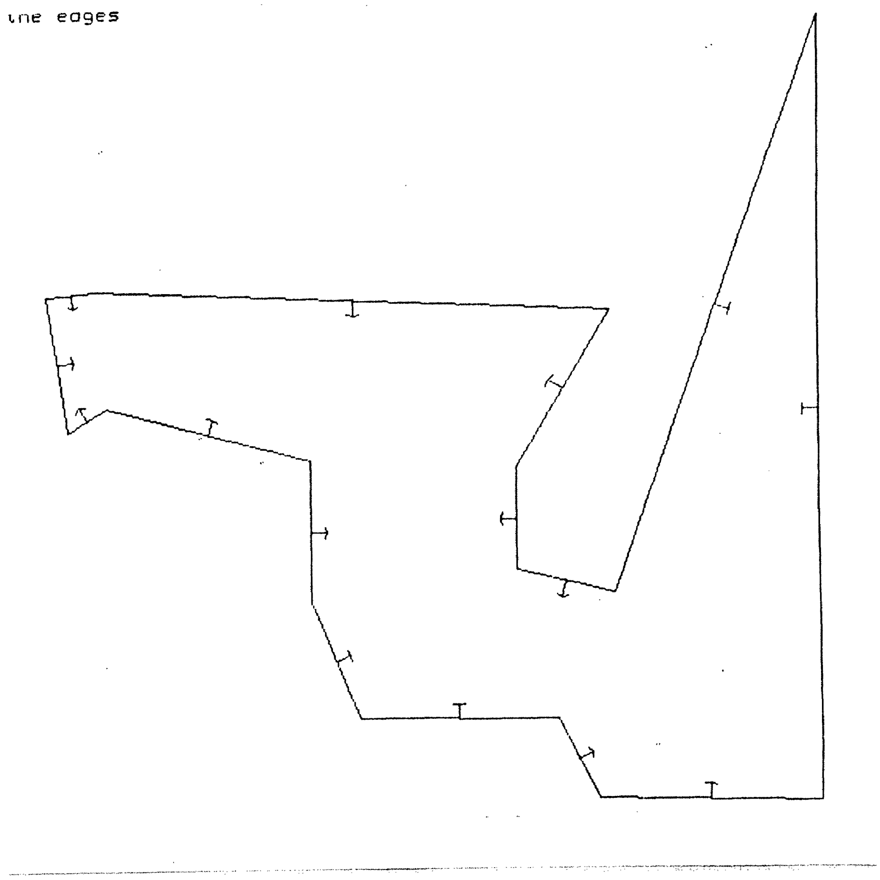

Figure 17. **A channel is hypothesized.**

7.3. Verifying a Path Using Configuration Space Techniques

The initial suggested path is a coarse-grained sequence of configurations within the channel volume. The next stage of the algorithm verifies these path nodes using configuration space techniques. Other path-verification techniques would also be possible.

Configuration Space as described by Lozano-Pérez [13, 14], Lozano-Pérez and Wesley, [15], represents the set of configurations an object can assume under translation and rotation. Thus for the two-dimensional mover's problem, configuration space is the Cartesian product of the two-dimensional plane $\Re^2$ with the one-dimensional sphere. Configurations that cause collisions with obstacles are *configuration obstacles* and form unreachable regions in the space. Thus in configuration space the obstacles are in effect expanded to fill the unreachable regions while the moving object is shrunk to a point.

To verify a configuration point along a suggested path, we must determine

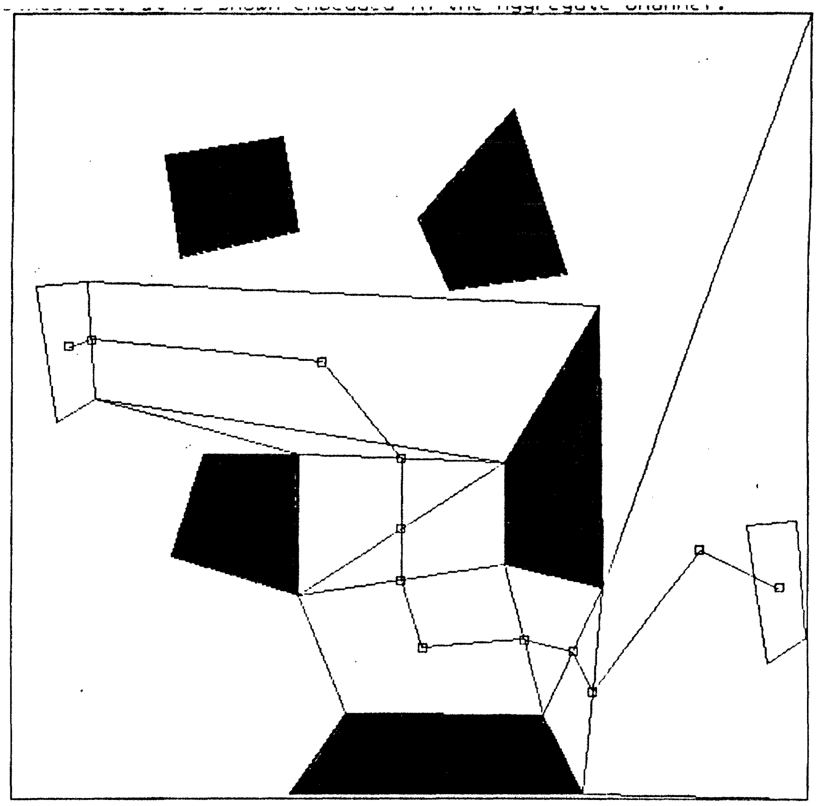

Figure 18. **A path is suggested through the channel.**

whether or not it is within a Configuration Space obstacle. To do this we use a module of the *C-Space* planner described and implemented by Brooks and Lozano-Pérez [5].

First, we identify all constraints on motion for the *C-Space* planner. In the *C-Space* method in two dimensions, only edges and convex vertices can generate motion constraints. Thus, our task amounts to taking the taking the outer union[12] of the channel volume and identifying all edges and convex vertices. In the *C-Space* literature these edges and vertices generate constraints on motion termed "type (b)" and "type (a)" respectively. Of these the type (a) vertex constraints are more complicated. The greatest reduction in constraint complexity in the channel transformation comes from the reduced number of convex vertices: this also coincides with our intuitive notion of a smooth, simply-connected channel volume.

[12] This is an easy operation since the channel path defines a non-overlapping partition of free space.

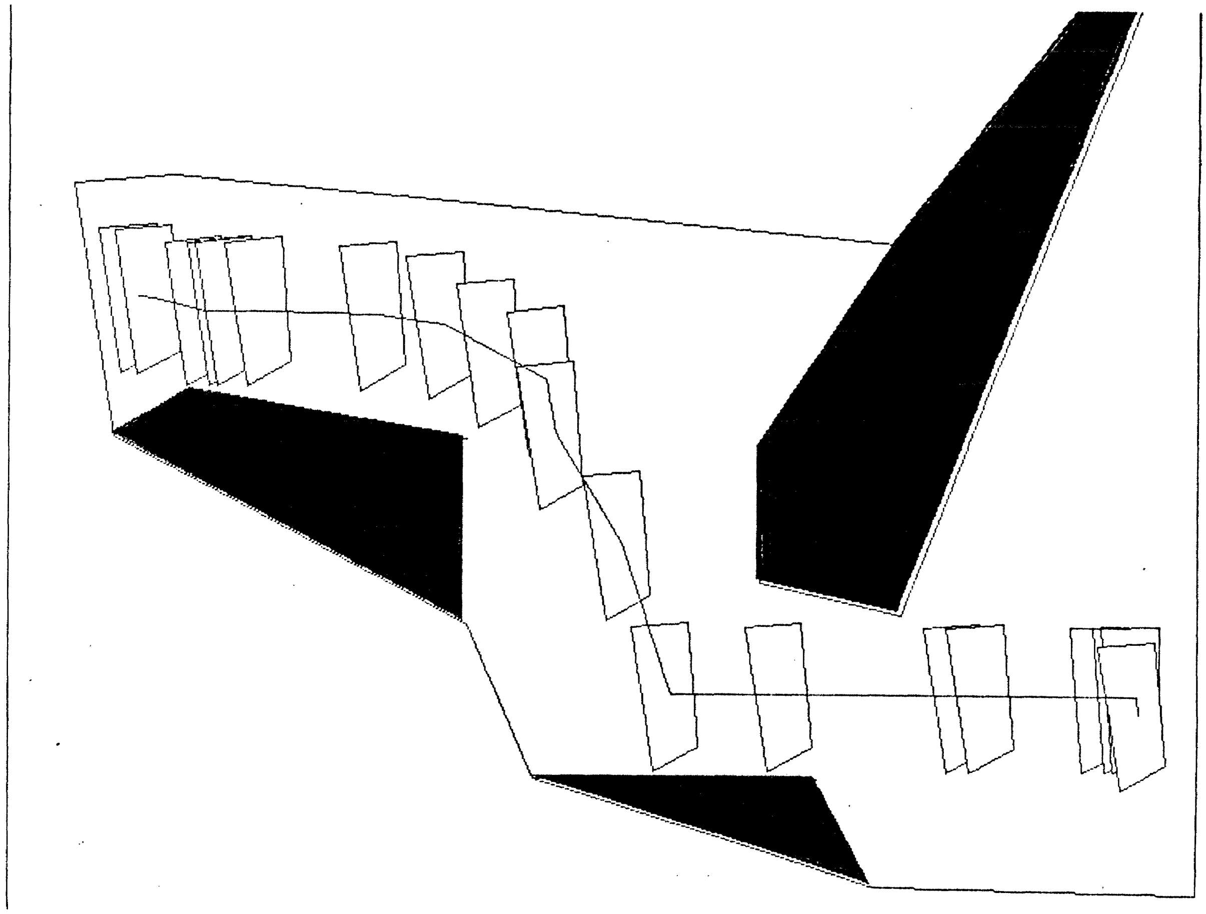

Figure 19. **The path is verified using *C-Space* techniques.**

Thus, we present the *C-Space* planner with a simplified workspace containing fewer geometric entities and generating fewer motion constraints. *C-Space* constraints are constructed only for those entities labeled as (a) or (b) generators.

7.4. Path Rectification and Interpolation

When a suggested configuration point along the path is found to be within a *C-Space* obstacle, it is frequently the case that the suggested node is valid within some other orientation with (x, y) held constant. Thus the channel path verifier attempts to "wriggle" in the θ dimension of configuration space. (It would also be possible to search in x and y). If no successful orientation can be found then the path node is abandoned. Otherwise, the new configuration with the rectified orientation is set up as a *sub-goal*, and backtracking occurs: two new subproblems are created in getting from the contiguous path nodes to the corrected configuration.

Once a suggested path is verified at the nodes, several successive *grains* of interpolation may be performed between verified configurations. The interpolation

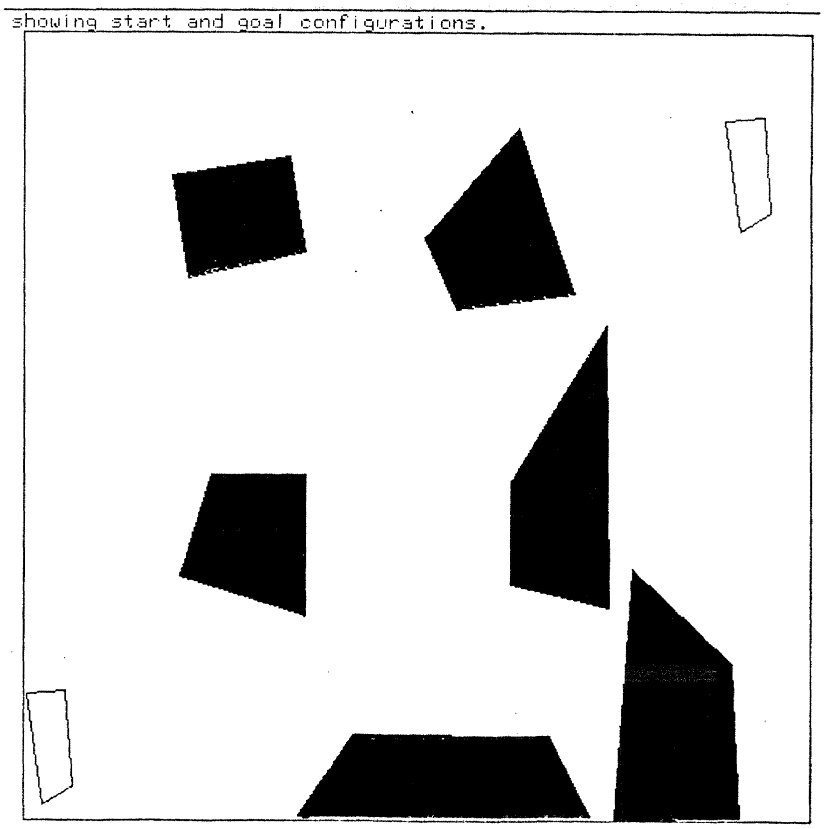

Figure 20. **A previously intractable problem that could be solved using channel construction with path hypothesis.**

suggests a sequence of configurations along a sub-path which is in turn verified at the nodes. Again, orientation rectification results in backtracking and reinterpolation from the preceding and subsequent configuration points. This interpolation can produce a smooth suggested path at any desired resolution.

7.5. Path Analysis and Homotopies: Introduction

Within a channel we hypothesize a paradigm path along centroids of volumes and free faces in the decomposition of free-space. This paradigm path is the axis of the homotopy cylinder for the path bundle. Corresponding to the union of channels there is the composition of homotopies:

$$\mathcal{K}_1 \bigcup \mathcal{K}_2 \qquad \text{(Channel Union)}$$

$$S \mapsto T \mapsto R \qquad \text{(Homotopy Composition)}$$

Further discussion of homotopies and channels can be found in section 8 and appendix II.

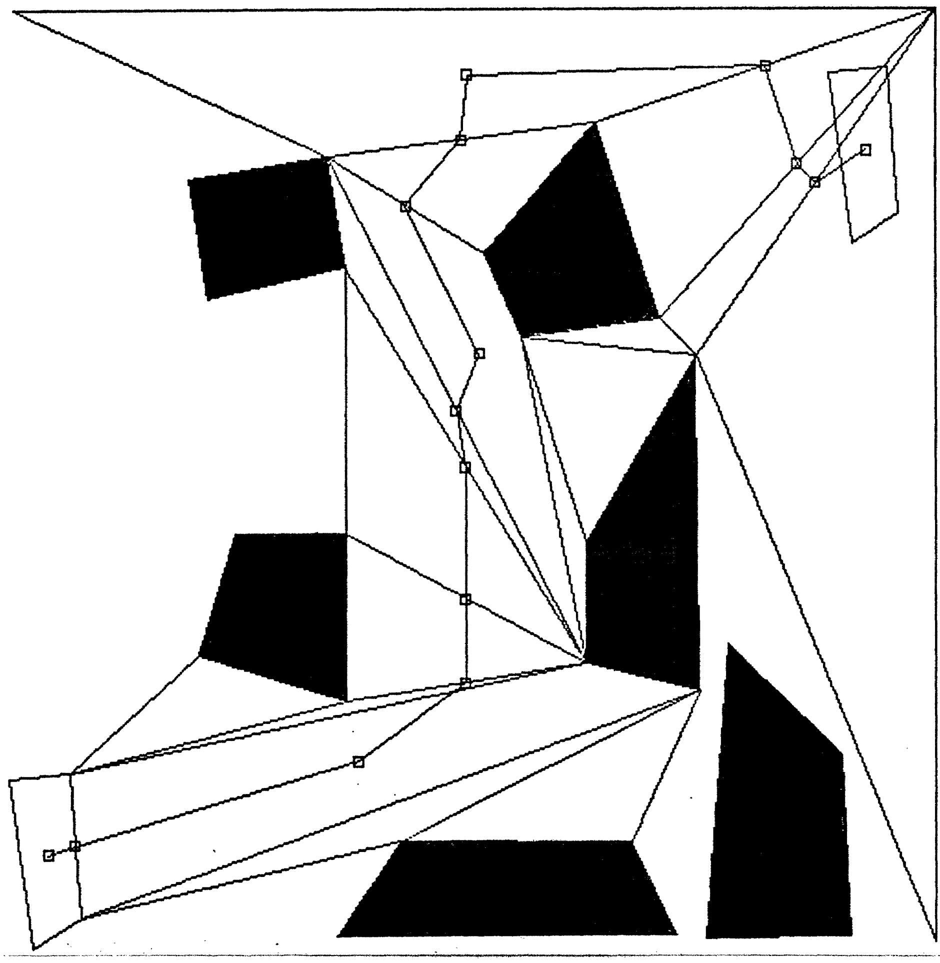

Figure 21. **Shows the multiply-connected channel decomposition from the constructive search, and a crude suggested path.**

8. Transformation to the Channel Domain

Our observation has been that in general, local algorithms can get lost examining irrelevant local constraints. A path planning algorithm starts with an hypothesis about a candidate path which is subsequently refined. Frequently this hypothesis is only implicit, for example, as a straight line connecting start and goal configurations or a search metric on the workspace. Without adequate global knowledge of the connectivity of the workspace and the classes of paths it contains, such methods may choose impossible or ill-advised candidate paths. In this section we present an intuitive analysis of how these paths may be "wrong" and why the mover's problem is easier in the channel domain. A more formal presentation

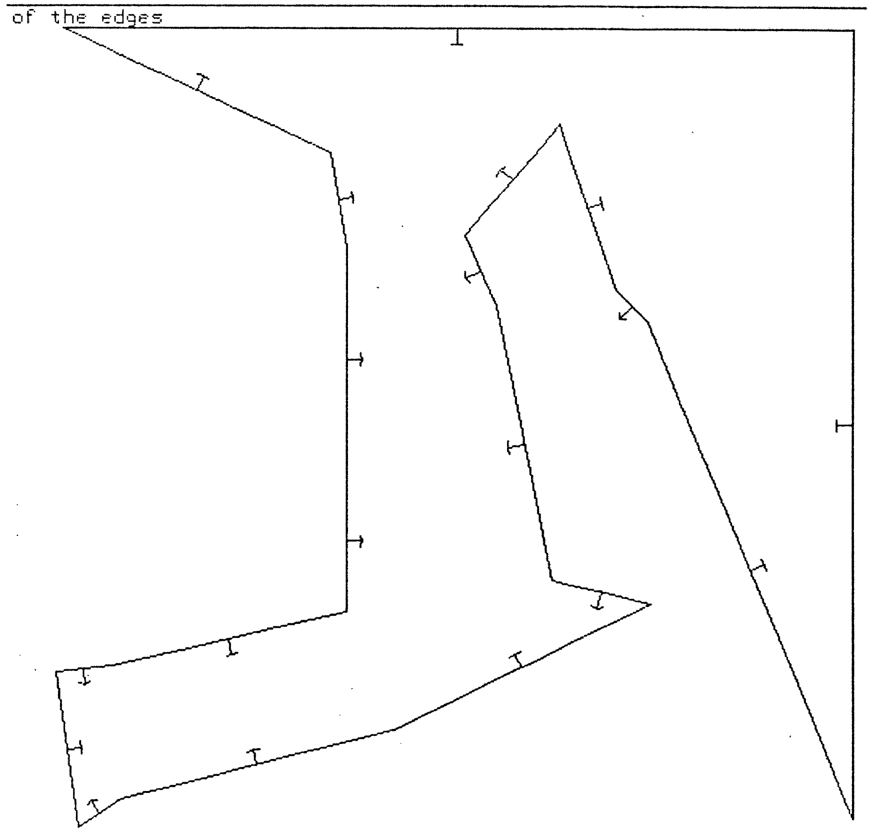

Figure 22. **Shows the outer boundary of the channel path.**

using homotopy theory (Hocking and Young, [12c]) can be found in an appendix. The formalization relies on the concept of equivalence classes of paths within the workspace. Intuitively, we say that there exists a continuous deformation between any two paths in the same equivalence class, and that the deformation does not leave the free-space.

8.1. Find-Path Subproblems and Straight-Line Approximations

Consider a successful path P in a workspace. For example, P might be an hypothesized path which is absolutely correct. Now consider a find-path subproblem Ξ on P, that is, the problem of getting from s to g where s and g are configurations on P. The better the straight-line approximation to each subproblem Ξ_i, the easier the global find-path problem.

P belongs to an equivalence class of paths which we denote by $[P]$. The

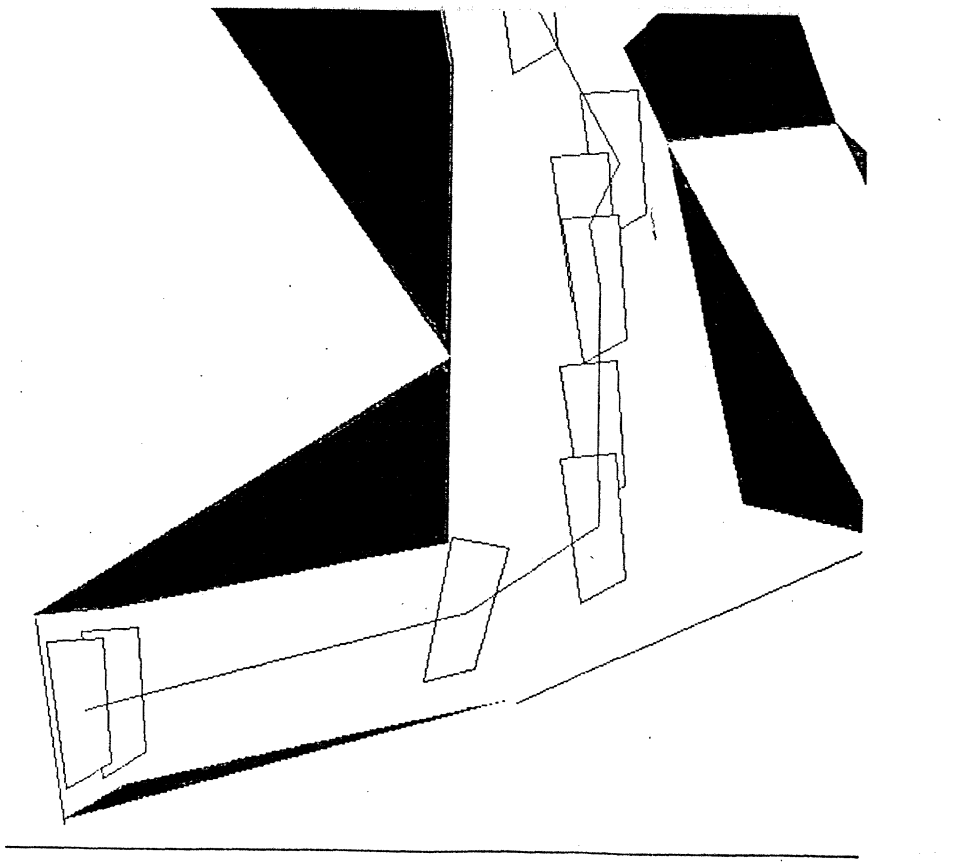

Figure 23a. **The *nodes* on the suggested path are rectified.**

workspace will also contain[13] classes of curves $[CO]$ entirely within obstacles. Finally, a multiply-connected workspace will contain a set of incorrect or inconsistent paths $[\wp]$. An incorrect class does not provide a path from the start to the goal. An inconsistent path is a member of an incompatible class which cannot be transformed into any path in $[P]$ without leaving the free-space. These path (and obstacle) classes induce an equivalence relation on configuration space. A straight-line approximation to any subproblem Ξ may contain configuration points in all three of these classes.[14]

$[CO]$ points lie within configuration obstacles and can be detected by the methods of Brooks and Lozano-Pérez [5]. The problem is that both $[P]$ points and $[\wp]$ points are in free-space and look the same to local methods. Within neighborhoods of $[\wp]$ points local progress may be made towards the goal. However,

[13]In this discussion we informally speak of a workspace or channel as "containing" a class of paths. In the appendix we will become more rigorous and deal with the space of functions whose images lie in a configuration space generated by the workspace obstacles.

[14]Of the classes in the taxonomy, only $[P]$ is an equivalence class. The other classes are more properly denoted by $[CO]^*$ and $[\wp]^*$.

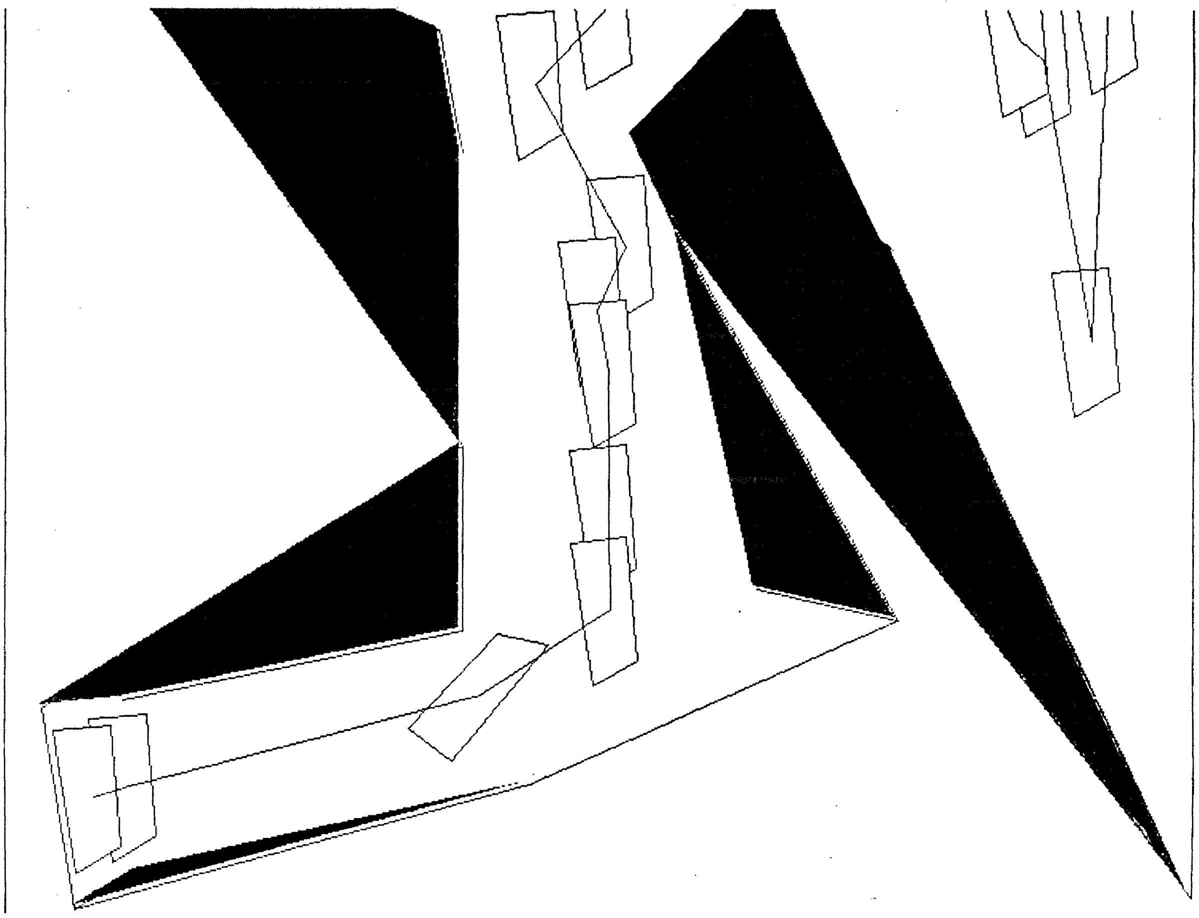

Figure 23b. **The result for a different rectification strategy.**

these candidate paths will eventually be blocked since the space of [℘] configurations is disconnected from the correct class of paths by configuration obstacles.

8.2. The Channel Transformation: Eliminating [℘]

[℘] points are the most troublesome, since they lie within free space and can be confused with path-correct $[P]$ points. The effect of the channel transformation is to rule out inconsistent or incorrect path classes within the transformed workspace, such that the straight-line approximation for any subproblem Ξ will contain only $[P]$ and $[CO]$ points. Within any such restricted subproblem, the $[CO]$ points may be detected and the path locally deformed into $[P]$.

This transformation is accomplished first through the construction of a simply-connected volume which can contain only one class of paths. If this volume is convex then the straight-line approximation for any subproblem will contain no [℘] points. In general the channel volume will not be convex, and thus the suggested path should form a visibility graph within the channel volume. That is, it should

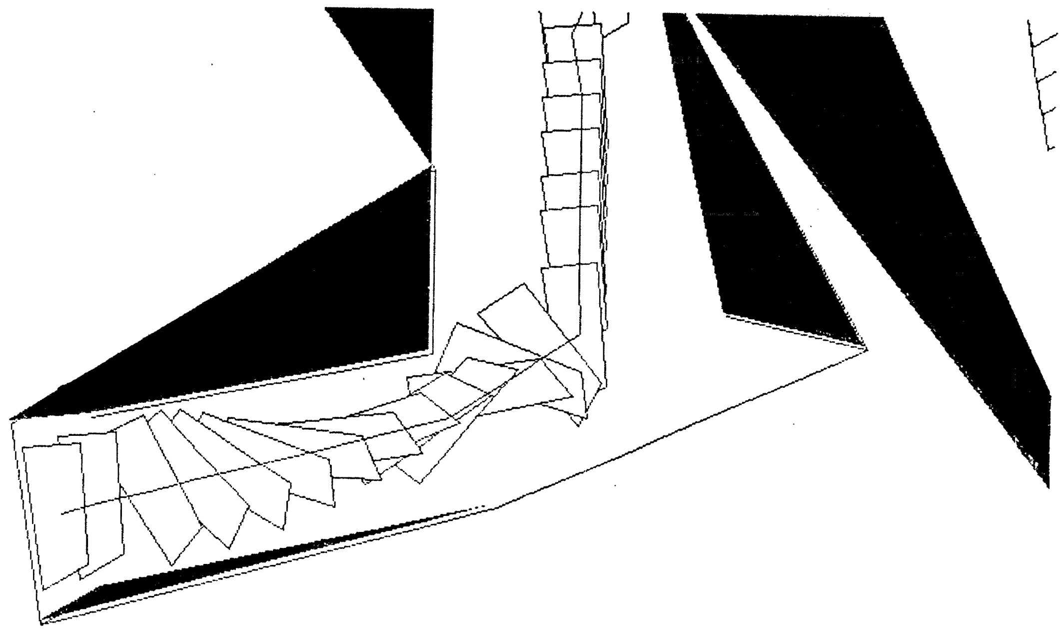

Figure 24. **A detail of path interpolation.**

partition the mover's problem into a set of subproblems in which (1) the start and goal configurations are known to be free and (2) the projections of the start and goal are mutually visible in the workspace. The subproblems in this partition are much easier find-path problems; a primary function of the path interpolation and field of view analysis is to ensure the visibility graph constraints.

8.3. What is a Reasonable Class of Paths?

The path equivalence classes admit paths which while topologically equivalent are clearly undesirable. These paths are those which are very long and stray very far from the set of minimal paths in the homotopy class. We can extend the above discussions to deal with reasonable and unreasonable paths in the same manner that we dealt with incorrect and inconsistent paths: consider a channel $\mathcal{K}$ containing a class of paths $[P_{\mathcal{K}}]$. Suppose that to wander outside of $\mathcal{K}$ means to take an unreasonable path. We thus wish to construct the subclass $[P_{\mathcal{K}}] \subseteq [P]$ to which we restrict all path hypotheses. The path class $[P]$ must be partitioned into the classes

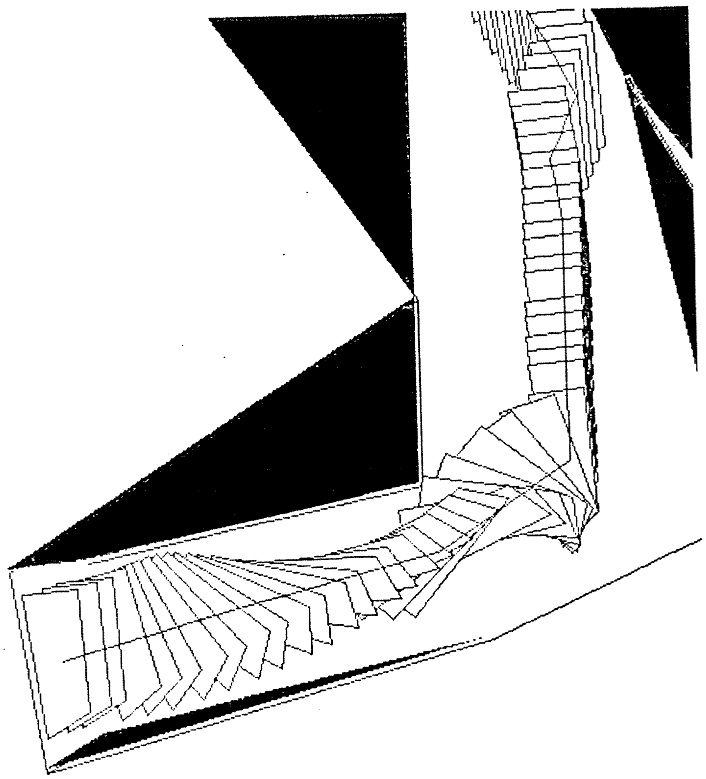

Figure 25. **A detail of path interpolation at a finer grain.**

$[P_K]$ and $[P] - [P_K]$, the latter corresponding to the unreasonable classes outside of K which will be placed in $[\wp]$. However, to enforce the partition of $[P]$ we must erect a barrier between $[P_K]$ and $[P] - [P_K]$. This barrier is generated precisely by the boundary of K. The reader is referred to the appendix for further details.

9. Future Research: Extensions to Higher Dimensions

We have attempted a general formulation of the channel transformation, while describing a two-dimensional implementation. Channel and path hypothesizing appear attractive as a technique for making the high-dimensional mover's problem more tractable. While much of the algorithm will extend directly for this future research, there are nevertheless unexplored areas and several major issues that must be addressed. These considerations include the following:

The geometric complexity will obviously be higher in three dimensions. The individual channel constructions can be performed in linear time. The convex hull

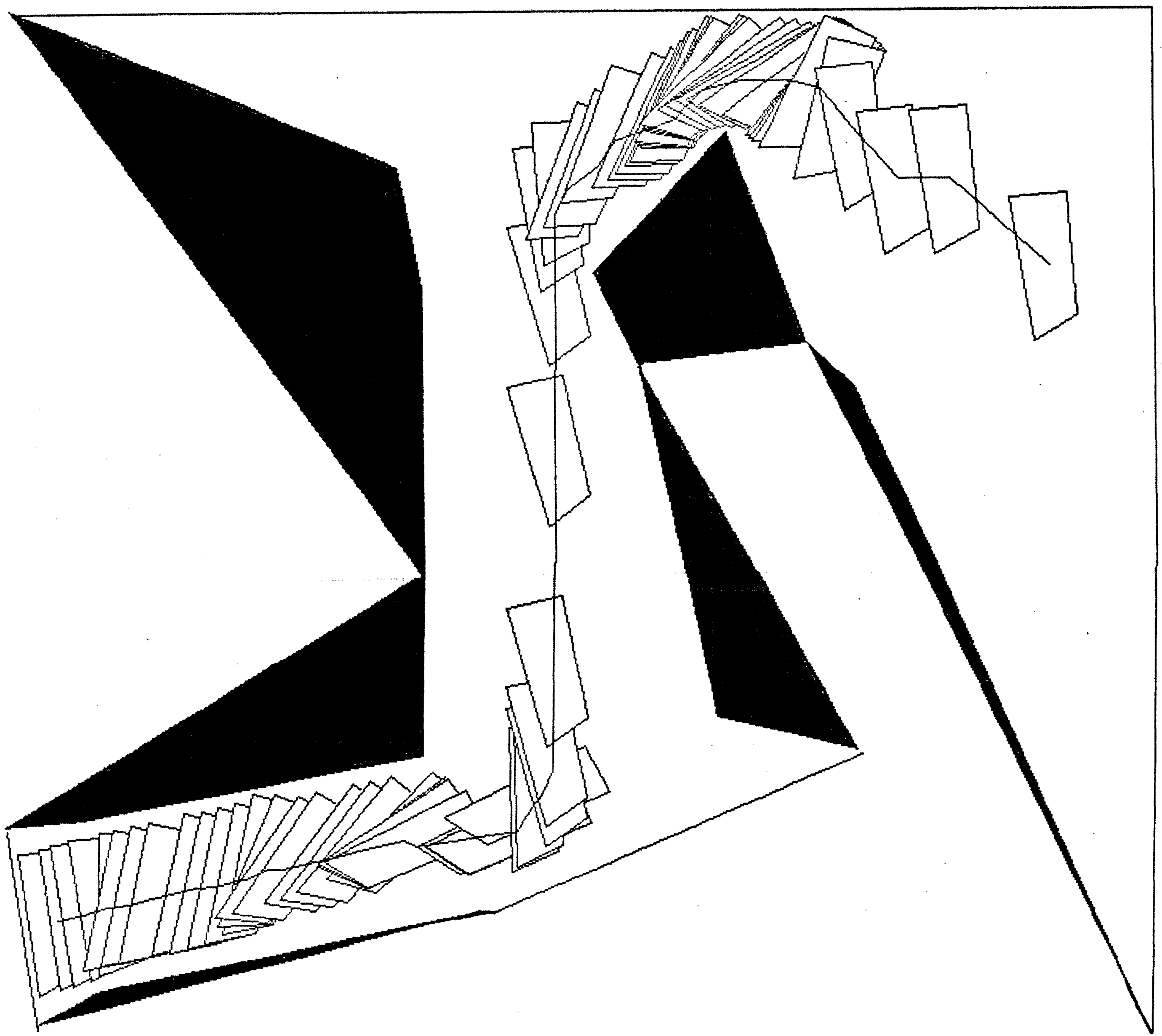

Figure 26. **The final path.**

and wedge construction operations have been implemented for the three-dimensional case (See figs. 28 and 29). The 3-dimensional visible-surface computation should be performed with the vantage point inside the frontier face, using a bounding rectangle around the face as the image plane. However, the visible surface calculation will be more complex, and in the current formulation must be performed from each frontier face prior to local construction. If the visible surface calculations contain

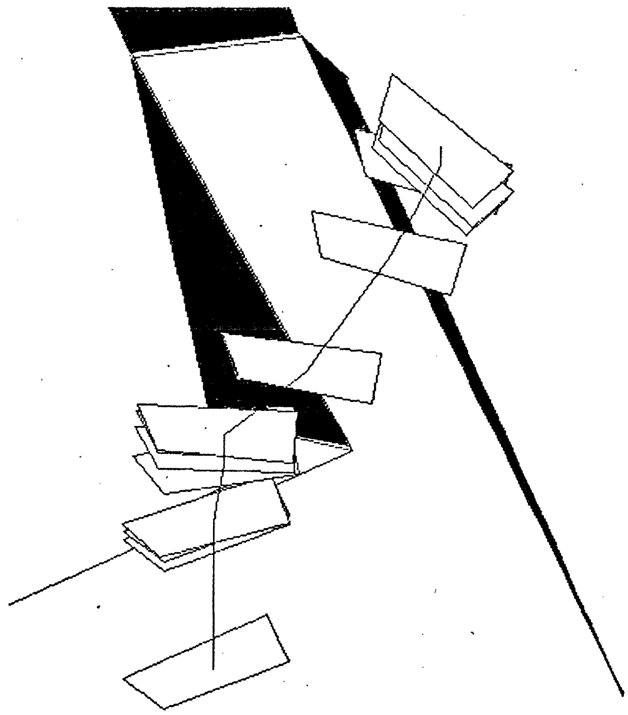

Figure 27. **A candidate path for a subproblem which crosses several path classes.**

steps of complexity $O(n \log n)$ (expected) to $O(n^2)$ (worst case), then the entire construction may be of $O(n^2 \log n)$ to $O(n^3)$. These are very rough estimates, since it is hard to obtain expected time estimates for algorithms which are sensitive to the particular workspace.

Fragmentation from partial construction will be more troublesome, as pointed out above. This requires that the field of view heuristics for selecting construction regions be more complicated. Construction must partition non-convex visible regions while minimizing fragmentation of containing regions. If fragmentation is unavoidable, then the resulting fragments should be convex if possible.

Finally, new motion constraint and path analysis techniques must be developed. Many of the same cross-section and size-thresholding methods may prove useful in path suggestion, but we ultimately need methods as strong as the *C-Space* verifier in higher dimensions.

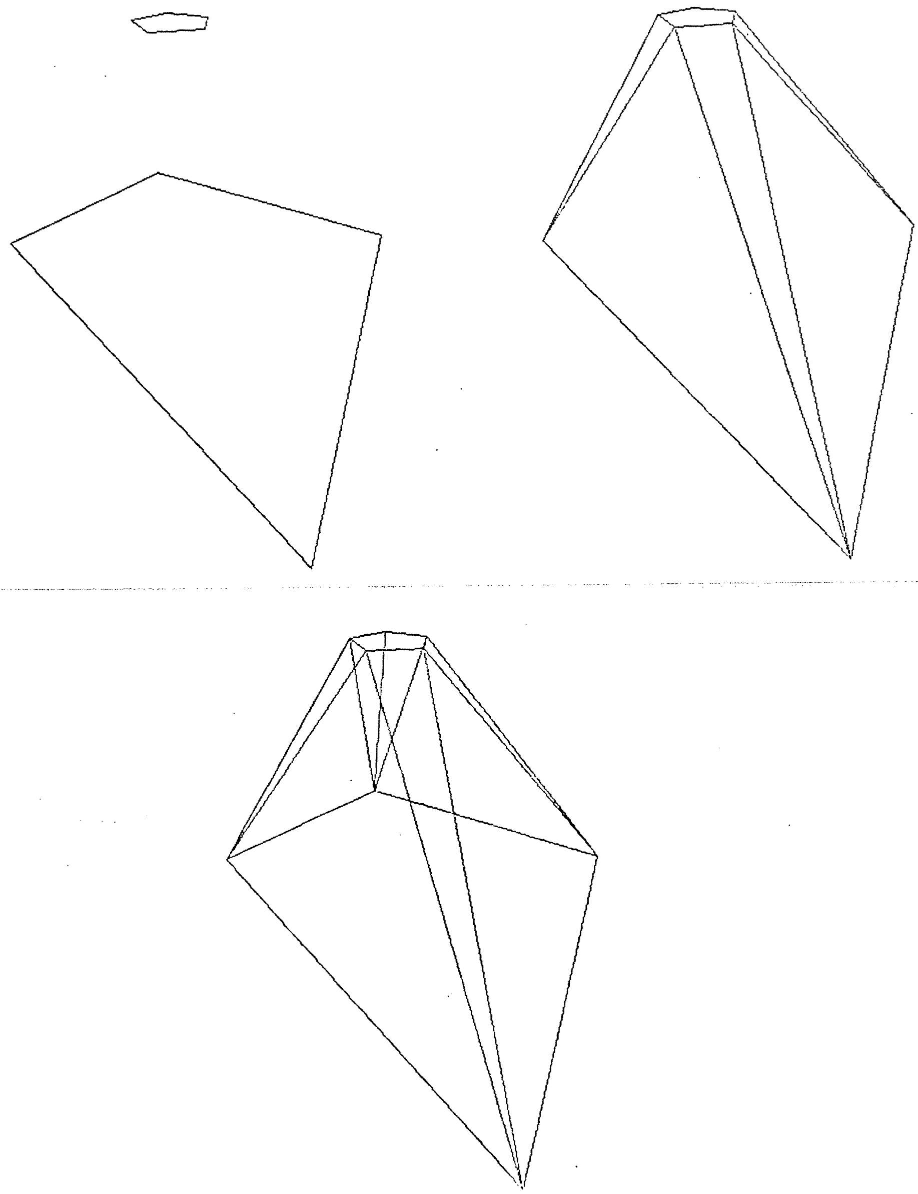

Figure 28. **Two obstacle faces viewed in perspective (a) and a channel $K = conv(vert(A) \bigcup vert(B))$ constructed between them (b), (c).**

10. Experiments and Results

The algorithm described here has been implemented for the two-dimensional case. The channel construction for complex workspaces takes on the order of a minute of "real" time as implemented on MIT and Symbolics Lisp Machines. Some of the hypothesized channels and paths are shown in figures (10-27).

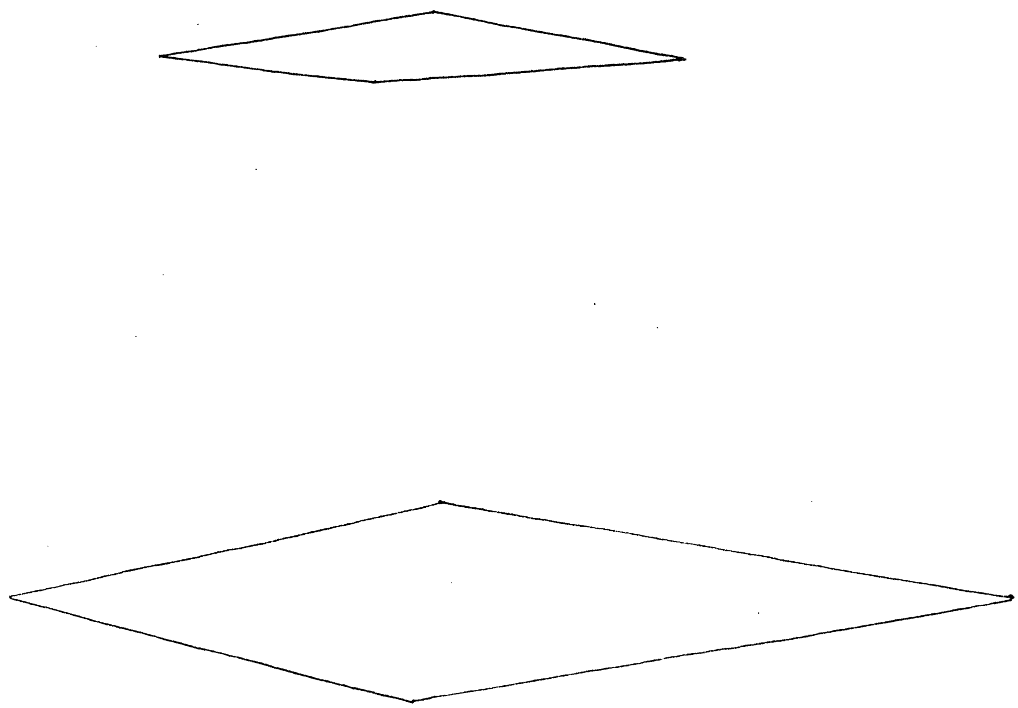

Figure 29a. **Two obstacle faces *A* and *B* viewed in perspective.**

The time for verification of a path using modules of the *C-Space* planner depends on the resolution and length of the path. The channel volume tends to have a constraint complexity which varies with the length of the path and which is relatively independent of the initial workspace. In addition, the constraint complexity tends to be significantly lower than in the initial workspace. The greatest reduction occurs in the number of type (a) constraints from convex vertices.

Although the channel path analysis can interpolate and verify a path to a given resolution, it cannot actually ensure that a free path exists between verified configurations. Of course for a fine-grained interpolation it is very unlikely that no path exists between the closely spaced "islands" along the path.

Since a chief deficiency of the *C-Space* cut-and-search algorithm lies in its "blindness," or inability to hypothesize reasonable paths and to set up good subgoals in complex workspaces, Brooks and Lozano-Pérez [5] suggest a *hybrid* approach whereby a likely path is suggested and the *C-Space* cut-and-search algorithm used

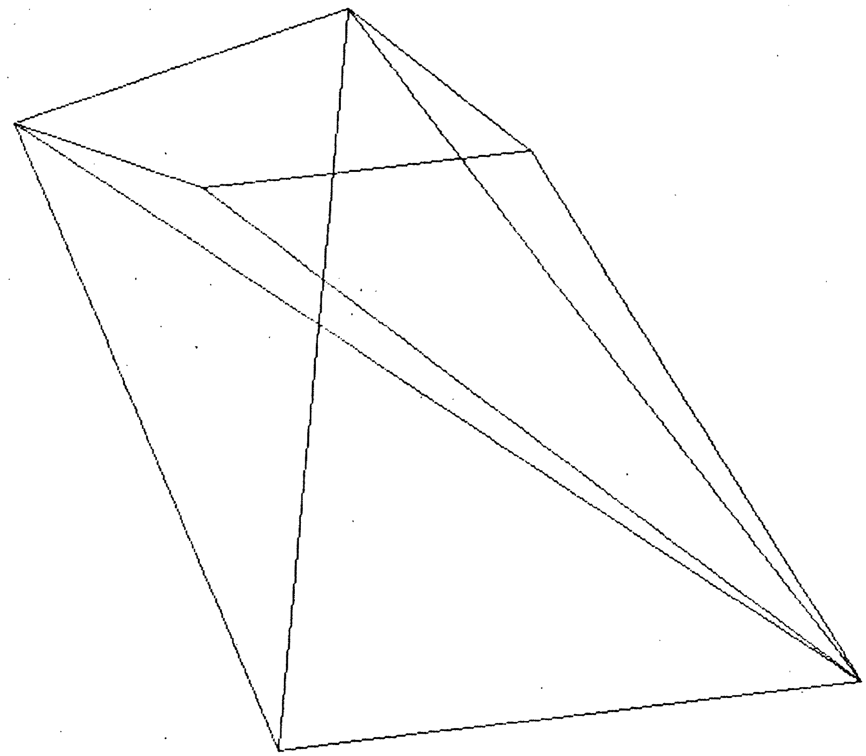

Figure 29b. **A wedge between A and an edge on B, showing back faces of the polyhedron.**

to verify and refine that path. Such an approach has the advantage that it is fully general in the sense that a path for an arbitrary object can be hypothesized and verified beyond any doubt. We have run experiments employing this method. The channel module constructed a channel volume of reduced geometric and motion constraint complexity. A path was suggested and interpolated at coarse-grained resolutions and the mover's problem partitioned into a sequence of subproblems in a visibility graph.

The transformed problem (the labeled channel volume) was "handed off" as a workspace to the *C-Space* cut-and-search planner. Finally, instead of allowing a "blind search" within the channel workspace, the *C-Space* planner was forced to use the interpolated and verified configuration path as a sequence of *planning islands* —in other words, to solve all the subproblems in the visibility graph partition of the global mover's problem.

We then compared the performance of the hybrid integrated channel and

C-Space system with that of the *C-Space* planner alone. The channel module constructed a channel workspace, hypothesized a path and partitioned the resulting find-path problem, and the *C-Space* module solved the sequence of simpler problems along the visibility graph partition. The result was a dramatic improvement in running times. In moderately complex workspaces the search was between two to ten times faster. More significant, however, is the fact that some previously intractable[15] problems for the *C-Space* planner can now be solved in total running times between 15 and 20 minutes. Figure (20) is such a case. This example was still not solved after 17 hours of running time by the *C-Space* planner alone; we estimate that even given unbounded resources a solution could not be found in under 48 hours. If a channel volume is hypothesized and the *C-Space* planner simply "turned loose" in the channel without path suggestion, a path was found in 7-plus hours. When the suggested path is partitioned into a visibility graph and the *C-Space* planner forced to solve the sequence of simpler find-path problems within the channel, the initial search for a path took 9 minutes and a final smoothing search took 5 more.

10.1. Related Approaches

The channel approach is an intuitive notion that has been appealed to in the literature: for example, Reif [21] uses the term channel to denote a slot through which an arm of a hinged body may slide. The method we have described is closely allied with Brooks' (Brooks, [4]) algorithm for "natural freeway" recognition using *generalized cones* (Binford, [2]) to represent the entire free space. Within each cone constraints on motion are derived. A path with rotations is found by intersecting constraints when transferring between cones. However it is not at all clear how to extend this technique to higher dimensions. An attempt has been made here to introduce a general channel formulation which can be extended to a three dimensional implementation.

The generalized cone find-path algorithm is sensitive to the geometric complexity of the environment, and less successful in workspaces littered with

[15] We use the term intractable in an empirical, and not a complexity-theoretic sense, to describe problems that take on the order of days to solve.

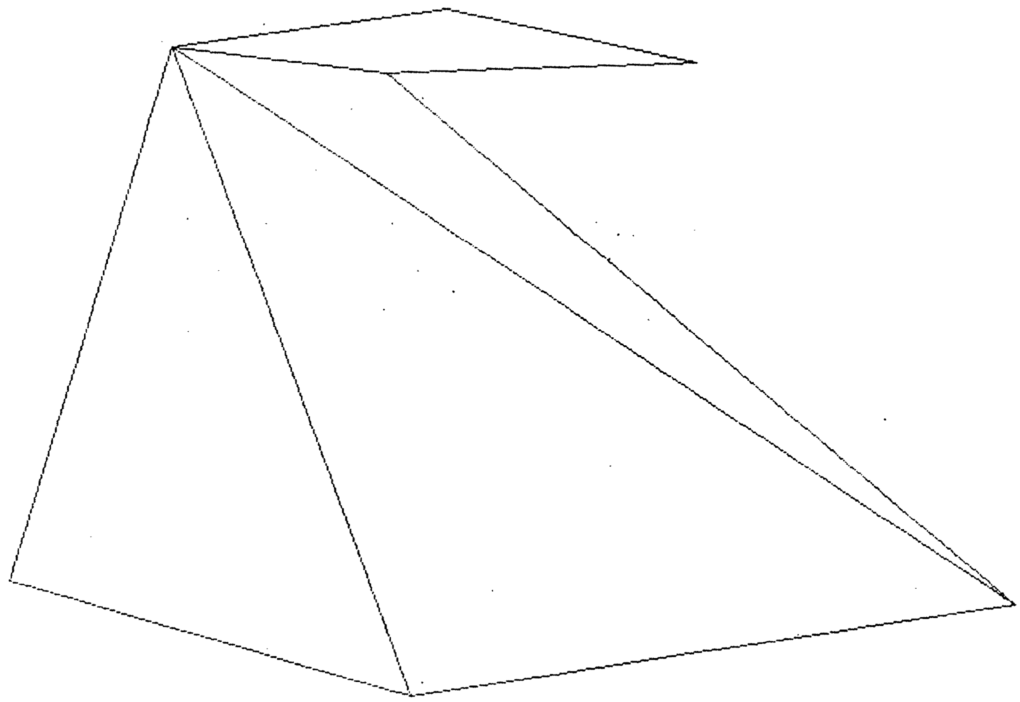

Figure 29c. **The implemented 3-D wedge constructor replaces the interior faces of the second wedge with those of the first wedge.**

many obstacles. The channel approach has not demonstrated this sensitivity since it attempts construction of a *simplified* environment of more or less uniform geometric and constraint complexity along a class of paths.

Cell decompositions have been used in other theories and implementations of spatial reasoning techniques. For example, see Forbus, [11], Lozano-Pérez [14], and Chatila, [6]. Freeway and channel partitions for characterizing the connectivity of free space are also related to *Voronoi diagrams* (Drysdale, [9]) which can be analyzed to find classes of paths (Ó'Dúnlaing and Yap, [18], Ó'Dúnlaing, Sharir and Yap [19]). However, Voronoi diagrams are difficult to construct in three dimensions. It has also been claimed that they exhibit extreme sensitivity to geometric variation; however, see Brady, [3a].

10.2. Improvements on the Method

It is possible for the heuristics to fail and for the hypothesized channel to be too narrow. To avoid this situation, we have adopted the heuristic of expanding the moving object slightly and then hypothesizing a channel for the "inflated piano." In addition some local expansion (construction) is performed near the start channel, since this region tends to be artificially narrow. The channel hypothesis is very conservative because of the cross-section thresholding: thus the homotopy of paths hypothesized is in no way the *minimum* class, but merely the "easiest". In a workspace where all path classes are extremely tight, the hypothesis is more susceptible to error.

Nevertheless, we have found that in general the channel volume is a good hypothesis for which a path-solution exists. The path suggestion heuristics are not as robust: there are cases where the path analysis can suggest difficult or unreasonable paths through legitimate channels. These typically arise for large, non-convex moving objects. The path suggestion heuristics currently employed are quite crude, and can be refined considerably. For example, "tight spots" within the channel could be identified by examining all interior free faces between obstacle surfaces. A narrow channel could be abandoned and a new channel path found or constructed, or the free space around tight spots could then be developed through local expansions. It is also possible to develop a more complicated constraint taxonomy, whereby constraints generated by obstacle faces and vertices would be given more weight than the artificial constraints from exterior free faces. There are a number of methodological and technical problems to be solved before this approach is feasible.

Alternatively, a measure of path-correctness could be calculated in the path verification stage. A poor measure of path-correctness could trigger backtracking and local constructive expansions around identified tight spots. Both the interior face analysis (above) and the location of failed path nodes could dictate the neighborhood of expansion. Workspaces and problems with poor path-correctness seem the best candidates for the hybrid channel and *C-Space* search method. For problems with a high measure of path-correctness at a fine-grain resolution the interpolated path should probably be taken as the final solution and the hybrid search forgone.

Perhaps the most encouraging result of this research is the dramatic reduction in the difficulty of the subproblems within the channel space. We believe that transformations of geometrically complex workspaces into simpler domains and partitions of spatial planning tasks into easier subproblems will play a key role in the future of geometric modeling for spatial reasoning.

Acknowledgments

I would like to thank Professor Tomás Lozano-Pérez for his guidance, advice, and detailed comments. Dr. Rodney Brooks provided comments and criticism on drafts of this paper and implementation of the integrated system. Dr. J. Michael Brady and Michael Erdmann read drafts, and discussed, reviewed, and aided the mathematical development. In addition, the following people gave generously of their time and made insightful suggestions: John Canny, Rich Doyle, and Professor John Hollerbach.

References

[1] **Baer, A., Eastman, C., and Henrion, M.** "Geometric Modeling: A survey," *Computer-Aided Design* **11**, 5 (1979).

[2] **Binford, Thomas** "Visual Perception by Computer," *IEEE Systems Science and Cybernetics Conference* , Miami, 1971.

[3] **Brady, J. M.** *et al. Robot Motion: Planning and Control* , MIT Press, Cambridge, MA, 1983.

[3a] **Brady, J. M.** "Criteria for Representations of Shape," *Human and Machine Vision* eds. Rosenfeld A., and Beck J., 1982.

[4] **Brooks, Rodney A.** "Solving the Find-Path Problem by Good Representation of Free Space," *IEEE Transactions on Systems, Man, and Cybernetics* SMC-**13** (1983).

[5] **Brooks, Rodney A. and Lozano-Pérez, Tomás** "A Subdivision Algorithm in Configuration Space for Findpath with Rotations," *International Joint Conference on Artificial Intelligence* , Karlsruhe, Germany, 1983.

[6] **Chatila, Raja** Systèm de Navigation pour un Robot Mobile Autonome: Modelisation et Processus Décisionnels, Ph.D. Thesis, L'Université Paul Sabatier de Toulouse, 1981.

[7] **Chazelle, Bernard** "Computational Geometry and Convexity," Department of Computer Science, Carnegie-Mellon University, CMU-CS-80-150, 1980.

[8] **Donald, Bruce R.** "The Mover's Problem in Automated Structural Design," *Proceedings, Harvard Computer Graphics Conference* , Cambridge, July, 1983.

[9] **Drysdale, Robert L.** Generalized Voronoi Digrams and Geometric Searching, Department of Computer Science, Stanford University, 1979.

[10] **Foley, J. D. and van Dam, A** *Principles of Interactive Computer Graphics* , Addison-Wesley, Reading, MA, 1982.

[11] **Forbus, Kenneth D.** "A Study of Qualitative and Geometric Knowledge in Reasoning about Motion," Massachusetts Institute of Technology Artificial Intelligence Laboratory, AI-TR-615, 1981.

[12a] **Giblin, P. J.** *Graphs, Surfaces, and Homology* , Chapman and Hall, London, 1977.

[12b] **Grünbaum** *Convex Polytopes* , Interscience Publishers, London, 1967.

[12c] **Hocking, J. and Young, G.** *Topology* , Addison-Wesley, Reading, MA, 1961.

[12d] **Kalay, Yehuda E.** "Determining the Spatial Containment of a Point in General Polyhedra," *Computer Graphics and Image Processing* **Vol. 19** (1982), 303–334.

[13] **Lozano-Pérez, Tomás** "Spatial Planning: A Configuration Space Approach," *IEEE Transactions on Computers* **C-32** (February, 1983).

[14] ___ "Automatic Planning of Manipulator Transfer Movements," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-11, No. 10** (1981).

[15] **___, and Wesley, M. A.** "An Algorithm for Planning Collision-Free Paths among Polyhedral Obstacles," *Communications of the ACM* **22**, 10 (1979).

[16] **Moravec, H. P.** "Visual Mapping by a Robot Rover," *Proceedings Sixth International Joint Conference on Artificial Intelligence* , Tokyo, Japan, 1979.

[16a] **Nievergelt J. and Preparata, F.** "Plane-Sweep Algorithms for Intersecting Geometric Figures," *Communications of the ACM* 25, 10 (1982).

[17] **Nilsson, Nils** *Principles of Artificial Intelligence* , Tioga Publishing Co., Palo-Alto, 1980.

[18] **Ó'Dúnlaing, C. and Yap, C.** "The Voronoi Diagram Method of Motion Planning: I. The Case of a Disc," Courant Institute of Mathematical Sciences, 1982.

[19] **Ó'Dúnlaing C., Sharir, M, C. and Yap, C.** "Retraction: A New Approach to Motion Planning," Courant Institute of Mathematical Sciences, 1982.

[20] **Preparata, F. and Hong, S.** "Convex Hulls of Finite Sets of Points in Two and Three Dimensions," *Communications of the ACM* **23**, 3 (1977).

[21] **Reif, John H.** "The Complexity of the Movers Problem and Generalizations," *Proceedings, 20^{th} Symposium on the Foundations of Computer Science* , 1979.

[22] **Requicha, A. A. G.** "Representation of Rigid Solids: Theory, Methods, and Systems," *ACM Computing Surveys* **12**, 4 (1980).

[23] **Schwartz, Jacob and Sharir, Micha** "On the Piano Movers Problem, II: General Techniques for Computing Topological Properties of Real Algebraic Manifolds," Courant Institute of Mathematical Sciences, Report No. 41, 1982.

[24] **Sechrest, Stuart and Greenberg, Donald** "A Visible Polygon Reconstruction Algorithm," *ACM Transactions on Graphics* **Vol. 1**, No. 1 (1982), 25-42.

[25] **Sutherland, Sproull,** *et al.* "A Characterization Of Ten Hidden-Surface Algorithms," *Acm Computing Surveys* **6**, 1 (1974).

[26] **Udupa,** S Collision Detection and Avoidance in Computer-Controlled Manipulators, Ph.D Thesis, Department of Department of Electrical Enginnering, California Institute of Technology, 1977.

[27] **Wittram, Martin** "A Hidden-Line Algorithm for Scenes of High Complexity," *IPC Business Press* **13**, 4 (1981).

Appendix I

In this appendix we present an algorithm for channel construction using wedges in 3-dimensions. Channel construction using pyramids is not addressed here, although the techniques are similar.

Let A, B, e_a, e_b, and $\mathcal{W}$ be as in section (4.1). Assume that the wedges $\mathcal{W}(A, e_b)$ and $\mathcal{W}(B, e_a)$ are both defined when we construct them. Our problem is (1) to construct the first wedge (2) to choose e_a correctly, (3) construct the second wedge (4) construct the outer union of the wedge complex. The outer union is computed by removing the wedge faces *interior* to the union and constructing a new polyhedron from the resulting face ring.

Let n_f denote the outward normal of a face f. We define the function $F_{int}(w, B)$ which computes which faces of the wedge $w = \mathcal{W}(A, e_b)$ will be *interior* to the channel polyhedron we construct as the union of two wedges. (There are two such interior faces). F_{int} uses the *reference face* B to make this determination, choosing $\{ f_1, f_2 \}$ on the boundary of w such that $n_{f_1} \cdot n_B$ and $n_{f_2} \cdot n_b$ are minimized.

To construct $\mathcal{K} = \mathcal{W}(A, e_b) \bigcup \mathcal{W}(B, e_a)$:

(1) Choose any $e_b \in \beta(B)$.[16]

(2) Construct $w_1 = \mathcal{W}(A, e_b)$.

(3) Compute the interior faces $\{ I_1, I_2 \} = F_{int}(w_1, B)$.

(4) Select e_a as follows:

$$e_a = \{ e \mid e \in \beta(I_1) \bigcup \beta(I_2) \text{ and } e \in \beta(A) \}$$

(5) Construct $w_2 = \mathcal{W}(B, e_a)$.

(6) Construct the channel polyhedron $\mathcal{K}$:

$$\mathcal{K} = \{ K \mid \beta(K) = \beta(w_1) \bigcup \beta(w_2) \ominus \{ I_1, I_2 \} \bigcup F_{int}(w_2, A) \}$$

[16] Typically the *largest* edge should be chosen.

Appendix II

Transformation to the Channel Domain

In this appendix we formalize the intuitive analysis of path classes presented in section (8). Our observation has been that in general, local algorithms can get lost examining local constraints. In particular, without adequate global knowledge of the connectivity of a workspace and the classes of paths it contains, such methods may choose impossible or ill-advised candidate paths. Here we examine in what way these paths may be "wrong," and how the mover's problem is easier in the transformed domain of the channel space.

11.3. Partitioning the Mover's Problem into Subproblems

Visibility Graph Constraints

The suggested path should form a visibility graph in the channel volume; that is, it should *partition* the mover's problem into a set of subproblems in which (1) the start and goal configurations are known to be free and (2) the $\Re^2$ projections of the start and goal are mutually visible in the workspace. The subproblems in this partition are much easier find-path problems; a primary function of the path interpolation and field of view analysis is to ensure the visibility graph constraints.

11.4. Channels and Homotopies

11.4.1. A Review of Elementary Homotopy Theory

There is a correspondence between channels and *homotopies* (see Hocking and Young, [12c] for a review). Let I^1 denote the unit interval. A parameterized family of mappings from a space X into a space Y is a continuous function $h : X \times I^1 \mapsto Y$. Consider the mappings f and g from X to Y: we say that h is a *homotopy* between f and g if for each point x in X,

$$h(x, 0) = f(x) \text{ and } h(x, 1) = g(x).$$

Intuitively the existence of h implies that f can be continuously deformed into g without leaving Y.

The homotopy relation between mappings from X into Y is an equivalence relation on the function space Y^X. Hence the homotopy relation partitions Y^X into disjoint equivalence classes, which are called *homotopy classes.* We write the homotopy relation as $f \simeq g$. These homotopy classes capture our intuitive notion of classes of paths. The homotopy classes of Y^X can be shown to be precisely the arcwise-connected components of Y^X (Hocking and Young, [12c]).

To take a concrete example, consider configuration space for the two-dimensional mover's problem to be the product space of the 2-dimensional Euclidean plane $\Re^2$ and the one-dimensional sphere S^1 to obtain $\Re^2 \times S^1$, and denote the configuration obstacles as $CO \subseteq \Re^2 \times S^1$. Now two paths f and g in the same equivalence class must belong to a parameterized family of mappings such that:

$$h : X \times I^1 \mapsto \Re^2 \times S^1 - CO.$$

and $h(x,0) = f(x)$, $h(x,1) = g(x)$ as before.

11.4.2. The Connectivity of Configuration Space

The configuration spaces $\Re^2 \times S^1$ (for the two-dimensional mover's problem) and $\Re^3 \times S^3$ (for the general three-dimensional case) are not simply-connected, since S^m is not simply-connected. The function space $(\Re^n \times S^m)^X$ contains several homotopy classes. For example, $(\Re^2 \times S^1)^X$ may be envisioned as a cylinder on which there are clearly two classes of paths: those that bound a 2-dimensional region and are contractable to a point, and those that go around the cylinder.

We would like to generate a configuration space which is simply-connected. Since this is not possible for the general product space $\Re^n \times S^m$ we will instead consider the product space of $\Re^n$ with the open intervals $[-\pi, \pi] \subseteq S^1$. Thus for the two-dimensional mover's problem we consider the product space

$$C = \Re^2 \times [-\pi, \pi].$$

In three dimensions this of course becomes $\Re^3 \times \Pi^3$ where

$$\Pi^m = [-\pi, \pi] \times \cdots \times [-\pi, \pi] \quad \text{(to } m\text{)}.$$

Π^m is homeomorphic to the interior of the m-cube. This new product space $\mathcal{C}$ is simply a restricted configuration space where the piano is not allowed to spin around wildly. Our motivation for constructing $\mathcal{C}$ is to show how channel constructions help transform problems in $\mathcal{C} - CO$ (which may be multiply-connected) into problems in a simply-connected *C-Space* containing one equivalence class of paths. In section (11.6.3) we sketch generalizations of these discussions to product spaces involving S^m in place of Π^m.

11.5. The Relation between Channels and Homotopies

A channel represents an hypothesis about a homotopy class. Consider

$$h : X \times I^1 \mapsto Y$$

.

Let $[f]$ be a homotopy class in Y^X. Thus $[f]$ is in the quotient space of Y^X induced by the homotopy equivalence relation: $[f] \in Y^X/\simeq$. We now induce an equivalence relation upon Y. Consider f_0, $f_1 \in [f]$, and their images on Y, the curves $C_0 = f_0(X)$ and $C_1 = f_1(X)$. Knowing that homotopy is an equivalence relation on the function space Y^X, we can decompose Y into equivalence classes that correspond to the images of the functions in the equivalence classes on Y^X. Thus $C_0 \simeq' C_1$ if, and only if $f_0 \simeq f_1$.

Now let Y be $\Re^2 \times \Pi^1 - CO = \mathcal{C} - CO$. Then $[f]$ is a *path-correct* homotopy class in that for all functions $f \in [f]$ the image of f in $\mathcal{C}$, $f(X)$, lies entirely outside the configuration obstacles (entirely within $\mathcal{C} - CO$). Without loss of generality, assume that the start and goal configuration are contained in each curve.[17] Also without loss of generality, assume that the reference point is contained within the moving object. (Any *C-Space* problem can be transformed so that this is the case. The reader will do best to imagine the reference point at the object's centroid for this discussion.)

[17] To insist that the start and goal configurations (s and g) are contained by all functions in an equivalence class is to consider homotopic equivalences *modulo* $s \bigcup g$, and the argument will be the same.

Now consider $\mathcal{H}_{[f]}$, the union of the image on Y of all functions $f_i \in [f]$:

$$\mathcal{H}_{[f]} = \bigcup_i f_i(X), f_i \in [f]$$

.

This is the region in Y which is covered by the class of paths $[f]$, and is the "maximal channel" in which every configuration point lies on a path in the same equivalence class. Note that since h is a continuous mapping, the equivalence class $[f]$ is uncountable and thus $\mathcal{H}_{[f]}$ is of course a union over an infinite number of paths. The fundamental correctness criterion for an hypothesized channel volume $\mathcal{K}$ is that it contain a projection into $\Re^2$ of a *slice* of correct paths in $\mathcal{C}$. This slice is parameterized by $I' \subseteq I^1, I' \neq \emptyset$, and we consider a subset of the parameterized family of mappings $h' : X \times I' \mapsto Y$. We denote this slice of paths as $[f]^{I'}$, and hence the criterion is:

$$\mathcal{H}_{[f]^{I'}} = \bigcup_{i \in I'} f_i(X), f_i \in [f]$$
$$Proj_{\Re^2}(\mathcal{H}_{[f]^{I'}}) \subseteq \mathcal{K}. \quad (1)$$

(1) is a necessary but not sufficient condition for channel-correctness, since while the projection of a configuration may be contained in the channel, we have not guaranteed that the object is contained in the channel at the required orientation. Assume a configuration has been found which was legal in the initial workspace but forbidden in the channel workspace. The only way this can happen is through the introduction of additional, artificial constraints (free faces). Thus the second fundamental criterion for channel-correctness ensures that the path class in the new domain is a subset of the initial path-correct homotopy class, $[f]$. $\mathcal{K}$ is the hypothesized channel in $\Re^2$, $i(\mathcal{K})$ its interior, and $\beta(\mathcal{K})$ its boundary. Let $CO_{\beta(\mathcal{K})}$ denote the set of configuration space obstacles generated by the channel boundary, and $CI_{i(\mathcal{K})} \subseteq \mathcal{C} - CO$ the set of configurations for which the piano is entirely within the channel. Now consider:

$$h' : X \times I' \mapsto CI_{i(\mathcal{K})}.$$

If h' exists, then the function space $CI^X_{i(\mathcal{K})}$ contains exactly one equivalence class, $[f_{\mathcal{K}}]$. Thus the second fundamental correctness criterion for channels is:

$$\begin{gathered} \exists h' : X \times I' \mapsto CI_{i(\mathcal{K})}, \\ [f_{\mathcal{K}}] \subseteq [f] \\ Proj_{\Re^2}(\mathcal{H}_{[f_{\mathcal{K}}]}) \subseteq Proj_{\Re^2}(\mathcal{H}_{[f]}). \end{gathered} \tag{2}$$

11.6. Homotopies, Channels, and Hardness

How is the mover's problem easier in the channel domain? In this section we address this issue by classifying the configurations on the straight-line approximation to the solution for a find-path subproblem. We first consider homotopy classes in channel space. Next we extend the discussion to general *C-Space*; and finally we discuss what it means to consider a "reasonable" class of paths.

11.6.1. Homotopy Classes in Channel Space

In this section we discuss in what way the mover's problem is easier in the channel domain. To facilitate this discussion we will speak of channels in the workspace as *corresponding* to homotopy classes in configuration space. Formally this implies the existence of a bijection between $\mathcal{K}$ and $[f_{\mathcal{K}}]$ where $[f]$, $\mathcal{K}$, and $[f_{\mathcal{K}}]$ satisfy (1) and (2) above. To see the correspondence, consider a complete partition of the workspace and the set of simply-connected channel paths.

We wish to consider three homotopy classes in channel space. The first is $[f_{\mathcal{K}}]$, the class of paths in $CI_{i(\mathcal{K})}$ that corresponds to the paths within the channel. Now, consider set of curves lying within configuration obstacles bounding the channel in $\mathcal{C}$, that is, the homotopies $h : X \times I^1 \mapsto CO_{\beta(\mathcal{K})}$. $CO_{\beta(\mathcal{K})}$ is homeomorphic to S^2 and the function space $CO^X_{\beta(\mathcal{K})}$ contains one equivalence class, $[CO]$. $[CO]$ is the second homotopy class. The third class contains paths incorrect or inconsistent with $[f_{\mathcal{K}}]$. An incorrect path does not provide a path from the start to the goal.

An inconsistent path is a member of an incompatible homotopy class such that no path in $[f_K]$ may be deformed into it without leaving the space $\mathcal{C} - CO$. We denote this last class by $[\wp]$. There is actually a fourth class of curves (those lying entirely within configuration obstacles not bounding the channel). Since these are clearly unreachable, we will classify them in $[\wp]$ also.

$$
\begin{aligned}
h_1 &: X \times I' \mapsto CI_{i(K)} && [f_K] \\
h_2 &: X \times I \mapsto CO_{\beta(K)} && [CO] \\
h_3 &: X \times I \mapsto \mathcal{C} - CO_{\beta(K)} - CI_{i(K)}. && [\wp]
\end{aligned}
$$

Now: consider pairwise unions of these classes. Paths in $[CO]$ can be continuously deformed into $[f_K]$ in the function space $(CI_{i(K)} + CO_{\beta(K)})^X$, and into $[\wp]$ in the function space $(\mathcal{C} - CI_{i(K)})^X$. In each case the union function space is simply the union of the component spaces; the resulting union space contains one arcwise-connected component and hence one equivalence class.[18] However, the union of $[f_K]$ and $[\wp]$ is disjoint, since the function space union $(\mathcal{C} - CO_{\beta(K)})^X$ contains two disjoint components separated by $CO^X_{\beta(K)}$. Hence the resulting union function space $(\mathcal{C} - CO_{\beta(K)})^X$ contains two homotopy classes: no homotopy spans $CI_{i(K)}$ and $(\mathcal{C} - CO_{\beta(K)} - CI_{i(K)})^X$ since they are disconnected by $CO^X_{\beta(K)}$.

A subproblem on a path P is a find-path problem Ξ from configuration s to g where s and g lie on P. Consider a successful path P in a complex workspace: for example, P might be an absolutely correct hypothesis about a path. Now for a subproblem Ξ, choosing s and g arbitrarily on P, there exists a path from the equivalence class $[f_K]$. However, consider the straight line connecting s and g. All points on the line lie on the image of functions in one of the three disjoint homotopy classes $[f_K]$, $[CO]$, or $[\wp]$. It is possible to formulate a dual taxonomy using $\mathcal{H}_{[f_K]}$, $\mathcal{H}_{[CO]}$, and $\mathcal{H}_{[\wp]}$ for points s' and g' in the initial workspace, which considers points in the correct channel, points on the obstacles bordering the channel, and points in an incorrect, inconsistent, or unreachable channels.

[18]Note that the function space $(\mathcal{C} - CI_{i(K)})^X$ is not simply-connected, and is homeomorphic to a filled 2-sphere with an internal cavity. However it is arcwise-connected, unlike $(\mathcal{C} - CO_{\beta(K)})^X$ which contains two disjoint components.

- The transformation to the channel domain essentially entails the elimination of case $[\wp]$ within each subproblem along the visibility graph of the suggested path. This is ensured by the simple-connectedness of the channel volume in the transformed workspace, and the visibility graph constraints on the local constructions and partitioning of the suggested path.

Paths containing configurations of types $[f_K]$ and $[CO]$ but not of type $[\wp]$ may be continuously deformed into the homotopy class $[f_K]$ in the simply-connected (union) function space $(CO_{\beta(K)} + CI_{i(K)})^X$ whose image in $\mathcal{C}$ is homeomorphic to the closure of a filled 2-sphere. This deformation if "off the obstacle and into the channel." However, no such continuous deformation exists in the function space for paths containing configurations of type $[\wp]$, which would require a deformation "out of the 'wrong' channel, through the obstacle, and into the 'right' channel." It should be intuitively clear that the latter rectification requires much greater topological changes in the path, and is thus much harder to effect. In particular, $[\wp]$ points are misleading: like $[f_K]$ points they lie in free-space and thus purely local methods cannot differentiate between path-correct and incorrect configurations. It is possible to make local progress in $[\wp]$ regions, yet these paths will eventually be found blocked by the disconnecting $[CO]$ region. We may think of type $[CO]$ configurations as *neighboring* the correct path class; the rectification of these collision points corresponds exactly to a continuous, local deformation of the path into $[f_K]$. Paths containing $[\wp]$ points cannot be so rectified since functions in incorrect or inconsistent homotopy class $[\wp]$ cannot be deformed into $[f_K]$. Thus we make the mover's problem easier by ruling out classes of topological impossibilities; this is done by ensuring that all intervening configurations in a subproblem lie either in the correct homotopy class or in the containing obstacle region.

11.6.2. Homotopy Classes in General *C-Space*

We now extend the above discussion to consider homotopy classes in general *C-Space* $(\Re^2 \times \Pi^1)$ with no channel and no visibility graph constraints on the partition of the mover's problem into subproblems. The basic result of this section will be that a subproblem Ξ is harder in this general *C-Space* since the straight-line approximation to the solution contains $[\wp]$ points. The previous argument was more intuitive, since the class $[\wp]$ was disconnected from the $[f_K]$ by an arcwise-connected space homeomorphic to the 2-sphere. Here the analysis is somewhat more difficult, since while the equivalence class of correct paths is still disconnected from the incorrect or inconsistent classes, the disconnecting region is no longer arcwise-connected (i.e., it may consist of several disjoint components).

Consider an equivalence class of correct paths $[f]$ in the function space $(\mathcal{C} - CO)^X$. Let $CO_{\beta(\mathcal{H}_{[f]})}$ denote the set of configuration obstacles bounding $\mathcal{H}_{[f]}$. $CO_{\beta(\mathcal{H}_{[f]})}$ may be a disconnected set, and thus the homotopy relation decomposes the function space $CO_{\beta(\mathcal{H}_{[f]})}{}^X$ into distinct equivalence classes, $[CO]^*$. In addition, we do not assume that $(\mathcal{C} - \mathcal{H}_{[f]} - CO_{\beta(\mathcal{H}_{[f]})})^X$ is arcwise connected, and allow it to have several homotopic equivalence classes of incorrect or inconsistent paths which we denote by $[\wp]^*$.

$$
\begin{aligned}
h_1 &: X \times I^1 \mapsto \mathcal{H}_{[f]} && [f] \\
h_2 &: X \times I^1 \mapsto CO_{\beta(\mathcal{H}_{[f]})} && [CO]^* \\
h_3 &: X \times I^1 \mapsto \mathcal{C} - \mathcal{H}_{[f]} - CO_{\beta(\mathcal{H}_{[f]})}. && [\wp]^*
\end{aligned}
$$

Now, consider subproblems such as Ξ in *C-Space* without a channel. With no visibility constraints on Ξ, configurations on the line (s, g) lie on images of functions in the distinct homotopy classes $[f]$, $[CO]^*$, or $[\wp]^*$. The crucial point is that path approximations containing $[\wp]^*$ configurations are disconnected from the correct homotopy class $[f]$ by some component of $[CO]^*$, and cannot be continuously deformed into $[f]$ in the function space union of $\mathcal{H}_{[f]}{}^X$ and $(\mathcal{C} - \mathcal{H}_{[f]} - CO_{\beta(\mathcal{H}_{[f]})})^X$.

11.6.3. The Product Space $\Re^2 \times S^1$

It is instructive to explore the complications introduced by the use of S^m in place of Π^m in construction of the product space. The effect is that more homotopy classes are formed in the function space. In this case channel construction can provide a reduction in the number of disjoint equivalence classes of paths, but not a reduction to a single class.

If we consider the configuration space $\Re^2 \times S^1$ instead of $\mathcal{C}$ then we know that $\Re^2 \times S^1$ is multiply-connected and the function space $(\Re^2 \times S^1)^X$ contains two homotopy classes. Then $CI_{i(\mathcal{K})}$ is now homeomorphic to the interior of the solid torus and $CO_{\beta(\mathcal{K})}$ to its boundary. $CI^X_{i(\mathcal{K})}$ contains two homotopy classes and $CO^X_{\beta(\mathcal{K})}$ three. These classes must be considered even in channel space, since any region in $\Re^2 \times S^1$ contains at least two path classes. The additional classes introduced by S^1 (or S^m) can be handled by again considering the equivalence classes $[CO]^*$ and factoring out homotopy classes of $[f_{\mathcal{K}}]^*$ or $[f]^*$ not homotopic to a constant mapping $c(X) = y_0$, where y_0 is some fixed point in $\Re^2 \times S^1$.

11.6.4. What is the Class of Reasonable Paths?

The homotopy equivalence classes admit paths which while topologically equivalent are clearly undesirable. These paths are those which are very long and stray very far from the set of minimal paths in the homotopy class. We can extend the above discussions to deal with reasonable and unreasonable paths in the same manner that we dealt with incorrect and inconsistent paths:

Note that $[f_{\mathcal{K}}] \subseteq [f]$ corresponds to the paths within the channel $\mathcal{K}$. Suppose that to wander outside of $\mathcal{K}$ means to take an unreasonable path. Then $[f_{\mathcal{K}}]$ corresponds to the reasonable, or shorter paths in $[f]$ that get to the goal *via* $\mathcal{K}$. By replacing $[f]$ by $[f_{\mathcal{K}}]$ in section (11.6.2) we can begin to talk about reasonable paths in the set $[f]$. This involves partitioning $[f]$ into the class of functions within the channel, $[f_{\mathcal{K}}]$ and the unreasonable class(es) $[f] - [f_{\mathcal{K}}]$, and placing $[f] - [f_{\mathcal{K}}]$ in $[\wp]^*$. However to enforce the partition of $[f]$ we must erect a barrier between $[f] - [f_{\mathcal{K}}]$ and $[f_{\mathcal{K}}]$. This barrier is precisely $CO^X_{\beta(\mathcal{K})}$, and is constructed by building the channel $\mathcal{K}$ in the workspace.