

On the Power of Manifold Samples in Exploring Configuration Spaces and the Dimensionality of Narrow Passages

Oren Salzman, Michael Hemmer, and Dan Halperin

Abstract—We extend our study of Motion Planning via Manifold Samples (MMS), a general algorithmic framework that combines geometric methods for the exact and complete analysis of low-dimensional configuration spaces with sampling-based approaches that are appropriate for higher dimensions. The framework explores the configuration space by taking samples that are *low-dimensional manifolds of the configuration space* capturing its connectivity much better than isolated point samples. The scheme is particularly suitable for applications in manufacturing, such as assembly planning, where typically motion planning needs to be carried out in very tight quarters. The contributions of this paper are as follows: (i) We present a recursive application of MMS in a six-dimensional configuration space, enabling the coordination of two polygonal robots translating and rotating amidst polygonal obstacles. In the adduced experiments for the more demanding test cases MMS clearly outperforms Probabilistic Roadmaps (PRM), with over 40-fold speedup in a six-dimensional *coordination-tight* setting. (ii) A probabilistic completeness proof for the case of MMS with samples that are affine subspaces. (iii) A closer examination of the test cases reveals that MMS has, in comparison to standard sampling-based algorithms, a significant advantage in scenarios containing *high-dimensional narrow passages*. This provokes a novel characterization of narrow passages, which attempts to capture their dimensionality, an attribute that had been (to a large extent) unattended in previous definitions.

Note to Practitioners—Highly constrained motion-planning scenarios, even of low degree of freedom, arise in various applications such as assembly planning and manufacturing applications. Our approach, which emphasizes high precision over any known sampling-based technique that we are aware of, allows to cope with exactly such cases. For instance, we show that our framework can be applied to tight scenarios that arise in three-handed assembly planning. The ability to cope with tight scenarios is possible, in part, due to recent improvements in exact geometric software such as the publicly available Computational Geometry Algorithms Library [43] (CGAL).

Manuscript received November 03, 2013; revised May 04, 2014; accepted June 12, 2014. Date of publication July 28, 2014; date of current version April 03, 2015. This paper was recommended for publication by Associate Editor M. A. Hsieh and Editor K. Lynch upon evaluation of the reviewers' comments. This work was supported in part by the Seventh Framework Program for Research of the European Commission under FET-Open Grant 255827 (CGL—Computational Geometry Learning), in part by the Israel Science Foundation under Grant 1102/11, and in part by the Hermann Minkowski—Minerva Center for Geometry at Tel Aviv University.

O. Salzman and D. Halperin are with the School for Computer Science, Tel-Aviv University, Tel Aviv 69978, Israel (e-mail: orensalz@post.tau.ac.il; danha@post.tau.ac.il).

M. Hemmer is with the Institute of Operating Systems and Computer Networks, University of Technology Braunschweig, 38106 Braunschweig, Germany (e-mail: hsaar@googlemail.com).

Digital Object Identifier 10.1109/TASE.2014.2331983

Index Terms—Robot motion planning, narrow passage, manifolds, probabilistic roadmaps (PRM), computational geometry algorithms library (CGAL).

I. INTRODUCTION

CONFIGURATION spaces, or C-spaces, are fundamental tools for studying a large variety of systems. A point in a d -dimensional C-space describes one state (or configuration) of a system governed by d parameters. C-spaces appear in diverse domains such as graphical animation, surgical planning, computational biology and computer games. For a general overview of the subject and its applications see, e.g., [11], [29], and [31]. The most typical and prevalent example are C-spaces describing mobile systems (“robots”) with d degrees of freedom (dofs) moving in some *workspace* amongst obstacles. As every point in the configuration space \mathcal{C} corresponds to a *free* or *forbidden* pose of the robot, \mathcal{C} decomposes into disjoint sets $\mathcal{C}_{\text{free}}$ and $\mathcal{C}_{\text{forb}}$, respectively. Thus, the motion-planning problem is commonly reduced to the problem of finding a path that is fully contained within $\mathcal{C}_{\text{free}}$.

A. Background

C-spaces for motion planning have been intensively studied for over three decades. Fundamentally, two major approaches exist.

1) *Analytic Solutions*: The theoretical foundations, such as the introduction of C-spaces [33] and the understanding that constructing a C-space is computationally hard with respect to the number of dofs [34], were already laid in the late 1970s and early 1980s in the context of motion planning. Exact analytic solutions to the general motion-planning problem as well as for various low-dimensional instances have been proposed in [5], [9], [10], [38] and [2], [3], [19], [33], [37], respectively. For a survey of related approaches, see [39]. However, only recent advances in applied aspects of computational geometry made robust implementations for important building blocks available. For instance, Minkowski sums, which allow the representation of the C-space of a translating robot, have robust and exact two- and three-dimensional implementations [16], [17], [45]. Likewise, implementations of planar arrangements¹ for curves ([43], C.30) [15], could be used as essential components in [38].

¹An arrangement of curves is a subdivision of the plane into zero-dimensional, one-dimensional and two-dimensional cells, called vertices, edges and faces, respectively, induced by the curves.

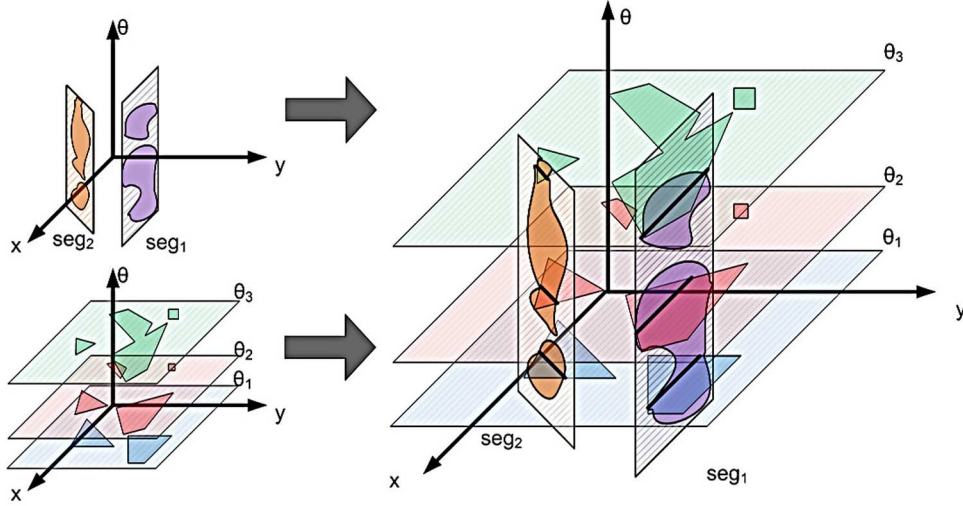


Fig. 1. MMS in three-dimensional C-spaces of translation and rotation in the plane. The left side illustrates two families of manifolds where the decomposed free cells are darkly shaded. The right side illustrates their intersection, which induces the graph \mathcal{G} . Figure taken from [36].

2) *Sampling-Based Approaches*: Sampling-based approaches, such as Probabilistic Roadmaps (PRM) [25], Expansive Space Trees (EST) [21] and Rapidly-exploring Random Trees (RRT) [30], as well as their many variants, aim to capture the connectivity of $\mathcal{C}_{\text{free}}$ in a graph data structure, via random sampling of configurations. For a general survey on the approach see [11] and [31]. As opposed to analytic solutions these approaches are also applicable to problems with a large number of dof. Importantly, the PRM and RRT algorithms were shown to be probabilistically complete [23], [27], [28], that is, they are guaranteed to find a valid solution, if one exists. However, the required running time for finding such a solution cannot be computed for new queries at runtime. This is especially problematic as these algorithms suffer from high sensitivity to the so-called “narrow passage” problem, e.g., where the robot is required to move in environments cluttered with obstacles, having low clearance.

Though there are also some hybrid approaches [14], [20], [32], [47] that incorporate both analytic and sampling-based approaches, it is apparent that the arsenal of currently available motion-planning algorithms lacks a general scheme applicable to high-dimensional problems with little or low sensitivity to narrow passages. In [36], we introduced a framework for Motion Planning via Manifold Samples (MMS), which also constitutes a hybrid approach. In a three-dimensional C-space, it was capable of achieving twenty-fold (and more) speedup factor in running time compared to the PRM algorithm when used for planning paths within narrow passages. We believe that the speedup presented in [36] does not present a mere algorithmic advantage for a specific implemented instance but a fundamental advantage of the framework when solving scenarios with narrow passages. The MMS framework is not the first to consider lower dimensional manifolds of the C-space. Several algorithms attempt to sample in the C-space, and project the sample to lower dimensional manifolds (see, e.g., [8] and [42]); however, these algorithms still sample points. For cases where some dimensions are presumed to be decoupled, such as

multirobot navigation, one can sample each robot's individual C-space (see, e.g., [4] and [44]) though these algorithms are typically not applicable when there is a tight coupling between the robots [41].

This study continues developing the MMS framework as a tool to overcome the gap mentioned in existing motion-planning algorithms. We briefly present the scheme and continue to a preliminary discussion on applying MMS in high-dimensional C-spaces, which motivates this paper.

B. Motion Planning via Manifold Samples

The framework is presented as a means to explore the *entire* C-space, or, in motion-planning terminology as a multi-query planner, consisting of a preprocessing stage and a query stage. The preprocessing stage constructs the *connectivity graph* \mathcal{G} of \mathcal{C} , a data structure that captures the connectivity of \mathcal{C} using *low-dimensional manifolds* as samples. The manifolds are decomposed into cells in $\mathcal{C}_{\text{free}}$ and $\mathcal{C}_{\text{forb}}$ in an analytic manner; we call a cell of the decomposed manifold that lies in $\mathcal{C}_{\text{free}}$ a *free space cell* (FSC). The FSCs serve as nodes in \mathcal{G} . Two nodes are connected by an edge if their corresponding FSCs intersect. See Fig. 1 for an illustration.

Once \mathcal{G} has been constructed it can be queried for paths between two configurations $q_s, q_t \in \mathcal{C}_{\text{free}}$ in the following manner: A manifold that contains q_s in one of its FSCs is generated and decomposed (similarly for q_t). These FSCs and their appropriate edges are added to \mathcal{G} . We compute a path γ (of FSCs) in \mathcal{G} between the FSCs that contain q_s and q_t . If such a path is found in \mathcal{G} , it can be (rather straightforwardly) transformed into a continuous path in $\mathcal{C}_{\text{free}}$ by planning a path within each FSC in γ .

C. MMS in Higher Dimensions

The successful application of MMS in [36] to a three-dimensional C-space can be misleading when we come to apply it to higher dimensions. The heart of the scheme is the choice of manifolds from which we sample. Informally, for the scheme to

work we must require that the used set of manifolds \mathcal{M} fulfills the following conditions.

C1—The manifolds in \mathcal{M} cover the C-space.

C2—A pair of surfaces chosen uniformly and independently² at random from \mathcal{M} intersect with significant probability.

C3—Manifolds need to be of very low dimension as MMS requires an analytic description of the C-space when restricted to a manifold. Otherwise, the machinery for the construction of this description is not readily available.

For MMS to work in C-spaces of dimension d , Condition **C2** has a prerequisite that the sum of dimensions of a pair of manifolds chosen uniformly and independently at random from \mathcal{M} is at least d with significant probability. This means, in particular, that \mathcal{M} will consist of manifolds of dimension³ $\lceil (d/2) \rceil$. With this prerequisite in mind, there is already much to gain from using our existing and strong machinery for analyzing two-dimensional manifolds [6], [7], [15], while fulfilling the conditions above: We can solve motion-planning problems with four degrees of freedom, at the strength level that MMS offers, which is higher than that of standard sampling-based tools.

However, we wish to advance to higher dimensional C-spaces in which satisfying all the above conditions at once is in general impossible. We next discuss two possible relaxations of the conditions above that can lead to effective extensions of MMS to higher dimensions.

Dependent Choice of Manifolds: If we insist on using only very low-dimensional manifolds even in higher dimensional C-spaces, then in order to guarantee that pairs of manifolds intersect, we need to impose some dependence between the choices of manifolds, i.e., relaxing condition **C2**. A natural way to impose intersections between manifolds is to adapt the framework of tree-based planners like RRT [30]. When we add a new manifold, we insist that it connects either directly or by a sequence of manifolds to the set of manifolds collected in the data structure (tree in the case of RRT) so far.

Approximating Manifolds of High Dimension: As we do not have the machinery to exactly analyze C-spaces restricted to manifolds of dimension three or higher, we suggest to substitute exact decomposition of the manifolds as induced by the C-space by some approximation. i.e., relaxing condition **C3**. There are various ways to carefully approximate C-spaces. In the rest of the paper, we take the approach of a *recursive application of MMS*.

In Section II, we demonstrate this recursive application for a specific problem in a six-dimensional configuration space, namely, the coordination of two planar polygonal robots translating and rotating amidst polygonal obstacles. In the adduced experiments for the more demanding test cases MMS clearly outperforms several variants and implementations of PRM with over 40-fold speedup in an especially tight setting.

²The requirement that the choices are independent stems from the way we prove completeness of the method. It is not necessarily an essential component of the method itself.

³The precise statement is somewhat more involved and does not contribute much to the informal discussion here. Roughly, \mathcal{M} should comprise manifolds of dimension $\lceil (d)/(2) \rceil$ or higher and possibly manifolds of their co-dimension.

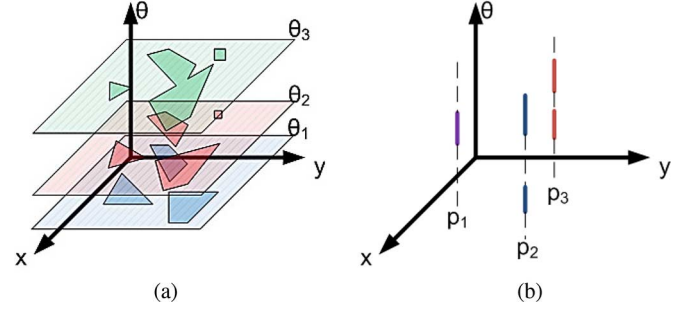


Fig. 2. Manifold families and their FSCs. FSCs of horizontal slices are polygons, while FSCs of vertical lines are intervals along the line. (a) Horizontal slices. (b) Vertical lines.

Section III provides the theoretical foundations for using MMS in a recursive fashion. In Section IV, we examine the significant advantage of MMS with respect to prevailing sampling-based approaches in scenarios containing *high-dimensional narrow passages*. This provokes a novel characterization of narrow passages, which attempts to capture their dimensionality. We conclude with an outlook on further work in Section V.

II. THE CASE OF TWO RIGID POLYGONAL ROBOTS

We discuss the MMS framework applied to the case of coordinating the motion of two polygonal robots R_a and R_b translating and rotating in the plane amidst polygonal obstacles. Each robot is described by the position of its reference point $r_a, r_b \in \mathbb{R}^2$ and the amount of counterclockwise rotation θ_a, θ_b with respect to an initial orientation. All placements of R_a in the workspace \mathcal{W} induce the three-dimensional space $\mathcal{C}^a = \mathbb{R}^2 \times S^1$. Similarly for R_b . We describe the full system by the six-dimensional C-space $\mathcal{C} = \mathcal{C}^a \times \mathcal{C}^b$.

A. Recursive Application of the MMS Framework

Had we had the means to decompose three-dimensional manifolds the application of MMS would be straightforward: The set \mathcal{M} consists of two families. An element of the first family of manifolds is defined by fixing R_b at free configurations $b \in \mathcal{C}_{\text{free}}^b$, while R_a moves freely inducing the three-dimensional subspaces⁴ $\mathcal{C}^a \times b$. The second family is defined symmetrically by fixing $a \in R_a$. As subspace pairs of the form $(a \times \mathcal{C}^b, \mathcal{C}^a \times b)$ intersect at the point (a, b) , manifolds of the two families intersect allowing for connections in the connectivity graph \mathcal{G} .

However, we do not have the tools to construct three-dimensional manifolds explicitly. Thus, the principal idea is to construct *approximations* of these manifolds by another application of MMS. Since for a certain manifold one robot is fixed, we are left with a three-dimensional C-space in which the fixed robot is regarded as an obstacle. Essentially, this is done by using the implementation presented in [36] but with a simpler set of manifolds (see also Fig. 2): (i) **horizontal slices**—corresponding to a fixed orientation of the moving robot while it is free to translate and (ii) **vertical lines**—corresponding to a fixed location of the reference point of the moving robot, while it is free to rotate.

⁴In this paper, when discussing subspaces, unless otherwise stated we refer to affine subspaces or linear manifolds.

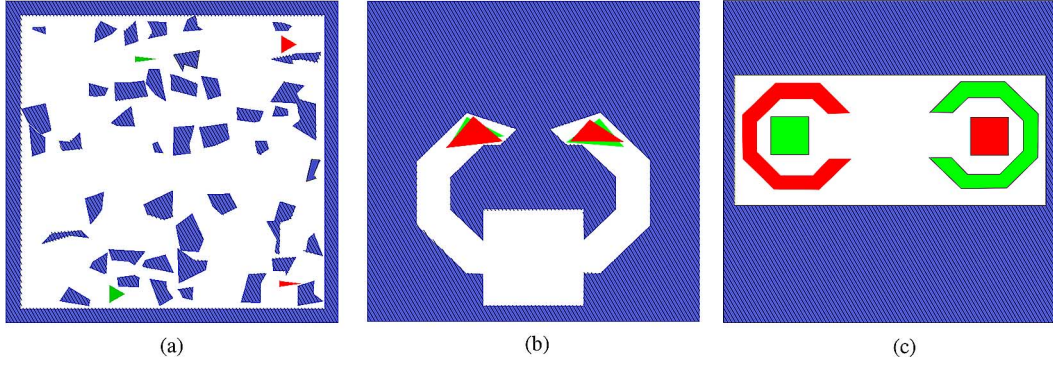


Fig. 3. Experimental scenarios. Source and target configurations are drawn in smoothly-textured green and red, respectively. Obstacles are drawn in striped-textured blue.

Since we only approximate the three-dimensional subspaces we have to make sure that they still intersect. In other words, if $\mathcal{C}_{\text{apx}}^a, \mathcal{C}_{\text{apx}}^b$ are the approximations of \mathcal{C}^a and \mathcal{C}^b , respectively, then $(a \times \mathcal{C}_{\text{apx}}^b, \mathcal{C}_{\text{apx}}^a \times b)$ intersect at the point (a, b) only if $a \in \mathcal{C}_{\text{apx}}^a$ and $b \in \mathcal{C}_{\text{apx}}^b$. To ensure this latter condition, we sample an initial set of angles Θ_a that is used for the first robot throughout the entire algorithm. When approximating its subspace (the second robot is fixed), we take a horizontal slice for each angle in Θ_a . At the same time, we only fix the robots position at angles in Θ_a . We do the same for the second robot and a set Θ_b . This way it is ensured that even the approximations of the three-dimensional subspaces intersect.

B. Implementation Details

Horizontal Slices: Let R_m and R_f denote the moving and fixed robot, respectively. Θ_m denotes the set of angles that is sampled for R_m . A horizontal plane for an angle $\theta_m \in \Theta_m$ is defined by the Minkowski sum of $-R_{\theta_m}$ with all the obstacles and, in addition, with the fixed robot.⁵ However, for each approximation of a three-dimensional affine subspace of the robot R_m , we are using the same set of angles,⁶ namely, Θ_m . Only the position of the robot R_f changes. Therefore, for all $\theta_m \in \Theta_m$ we precompute the Minkowski sum of $-R_{\theta_m}$ with all the obstacles. In order to obtain a concrete slice we only need to add the Minkowski sum of $-R_{\theta_m}$ with R_f . This can be done by a simple overlay operation (see, e.g., [15, C.6]).

Vertical Lines: Fixing the reference point of R^m to some location while it is free to rotate induces a vertical line in the three-dimensional C-space. Each vertex (or edge) of the robot in combination with each edge (or vertex) of an obstacle (or the fixed robot) give rise to up to two critical angles on this line. These critical values mark a potential transition between $\mathcal{C}_{\text{forb}}$ and $\mathcal{C}_{\text{free}}$. Thus, a vertical line is constructed by computing these critical angles and the FSCs are maximal free intervals along this line; for further details see [35].

⁵ $-R_{\theta_m}$ denotes R_m rotated around the origin by θ_m and reflected about the origin.

⁶We note that in our implementation, we add a random shift to the set of slices to avoid situations where the initial configuration of one of the robots is aligned with a narrow passage [as is the case in Fig. 3(c)]. This is done for each robot independently.

C. Experimental Results

We demonstrate the performance of our planner using three different scenarios in six-dimensional C-spaces. All scenarios consist of a workspace, obstacles, two robots, and one query (source and target configurations). Fig. 3 illustrates the scenarios where the obstacles are drawn in blue and the source and target configurations are drawn in green and red, respectively. All reported tests were measured on a Lenovo T420 with a 2.8 GHz Intel Core i7-2640M CPU processor and 8 GB of memory running with a Windows 7 64-bit OS. Preprocessing times are the average of 12 runs excluding the minimal and maximal values. The algorithm is implemented in C++ based on CGAL [43] and the BOOST Graph Library [40], which are used for the geometric primitives, and the connectivity graph \mathcal{G} , respectively.

We chose to compare our planner to the implementation of PRM provided by OMPL [12]. In addition, we also compare with Obstacle-Based PRM (OB-PRM) [1] and Uniformly distributed Obstacle-Based PRM (U-OB-PRM) [48] (also implemented in OMPL), which were shown to perform better than PRM in many scenarios where narrow passages exist. We manually optimized the parameters of each planner over a concrete set. The parameters used by MMS are: n_θ —the number of sampled angles; n_ℓ —the number of vertical lines; n_f —the number of times some robot is fixed to a certain configuration while the three-dimensional C-space of the other is computed. The parameters used for the PRM algorithms are: k —the number of neighbors to which each milestone should be connected; res—collision-checking resolution. U-OB-PRM needs additional parameters, the length l of the line-segments sampled in space and the resolution of samples along this line. Following the results of [48] and after validating these parameters, we used the same collision checking-resolution for the resolution and a line-segment of length equal to ten times the collision-checking resolution. We found empirically that in order to obtain the best results from U-OB-PRM, we should add uniform samples to the biased ones. Thus, the variant we used samples half of the time uniformly in space, while half of the time uses the scheme suggested in [48]. Table I summarized the parameters used by each algorithm, the average running time and the standard deviation (denoted by t and stdev, respectively).

TABLE I
COMPARISON OF MMS WITH PRM VARIANTS

Scenario	MMS					PRM				OB-PRM				U-OB-PRM			
	n_θ	n_ℓ	n_f	t[sec]	stdev	k	res	t[sec]	stdev	k	res	t[sec]	stdev	k	res	t[sec]	stdev
Random polygons	5	512	2	8	1.6	10	0.02	14.5	8.3	10	0.01	28.4	12	8	0.01	10.5	9.9
Viking Helmet	20	16	10	6.2	1.2	10	0.005	86.8	34	10	0.005	92.8	14	8	0.0125	40	28
Pacman	5	4	180	17.6	3.5	12	0.015	15	9.5	10	0.01	18.7	6.8	10	0.0125	20	3.3

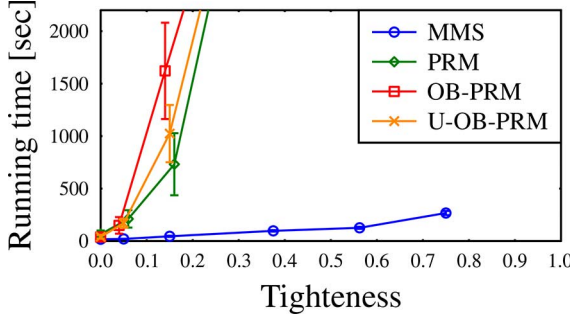


Fig. 4. Tightness Results. Error bars represent one standard variation.

The Random polygons scenario⁷ is an easy scenario where little coordination is required. All planners require roughly the same amount of time to solve this case. We see that even though our planner uses complex primitives, when using the right parameters, it can handle simple cases with no overhead when compared to the PRM algorithms.

The Viking-helmet scenario consists of two narrow passages that each robot needs to pass through. Moreover, coordination is required for the two robots to exchange places in the lower chamber. We see that the running times of the MMS implementation are favorable when compared to the PRM implementations. Note that although each robot is required to move along a narrow passage, the motion along this passage does not require coordination between the robots.

The Pacman scenario, in which the two robots need to exchange places, requires coordination of the robots: they are required to move into a position where the C-shaped robot, or Pacman, “swallows” the square robot; the Pacman is then required to rotate around the robot. Finally, the two robots should move apart (see Fig. 5). We ran this scenario several times, progressively increasing the square robot size. This caused a “tightening” of the passages containing the desired path. Fig. 4 demonstrates the preprocessing time as a function of the tightness of the problem for both planners. A tightness of zero denotes the base scenario [Fig. 3(c)], while a tightness of one denotes the tightest solvable case. Our algorithm is less sensitive to the tightness of the problem when compared to the PRM algorithm. In the tightest experiment solved by all PRM variants, MMS runs ten times faster. We ran the experiment on tighter scenarios but all PRM algorithms crashed after 5000 s due to lack of memory resources. We believe that the behavior of the algorithms with respect to the tightness of the passage reveals a fundamental difference between the two algorithms and discuss this in Section IV.

⁷A scenario provided as part of the OMPL distribution.

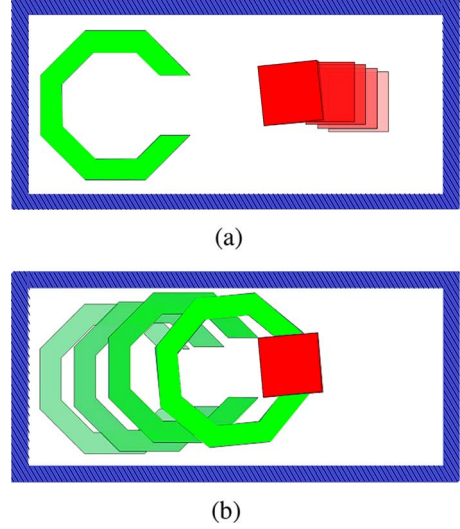


Fig. 5. Example of a path in the Pacman Scenario. (a) The square robot moves into a position where the Pacman can engulf it. (b) The Pacman engulfs the square robot.

III. PROBABILISTIC COMPLETENESS OF MMS

An algorithm is *probabilistically complete* if the probability that it will produce a solution (when one exists) approaches one as more time is spent. It has been shown that PRM, using point samples, is probabilistically complete (see, e.g., [11, C.7]). At first glance, it may seem that if the scheme is complete for point samples then it is evidently complete when these samples are substituted with manifold samples: manifolds of dimension one or higher guarantee better coverage of the configuration space. However, there is a crucial difference between PRM and MMS when it comes to connectivity. The completeness proof for PRM relies, among others, on the fact that if the straight line segment in the configuration space connecting two nearby samples lies in the free space, then the nodes corresponding to these two configurations are connected by an edge in the roadmap graph. The connectivity in MMS is attained through intersections of manifolds, which may require a chain of subpaths on several distinct manifolds to connect two nearby free configurations. This is what makes the completeness proof for MMS nontrivial and is expressed in Lemma III.3.

We present a probabilistic-completeness proof for the MMS framework for the case where the C-space \mathcal{C} is the d -dimensional Euclidean space \mathbb{R}^d , while MMS is taking samples from two perpendicular affine subspaces, the sum of dimensions of which is d . Assuming that the C-space is Euclidean does not impose a real restriction as long as the actual C-space can be embedded in a Euclidean space (see, e.g., ([11, Secs. 3.5 and 7.1.2], ([31, Ch. 4–5] or [26])).

Let A and B denote affine subspaces of \mathcal{C} and let k and $d - k$ be their dimensions, respectively. As \mathcal{C} is

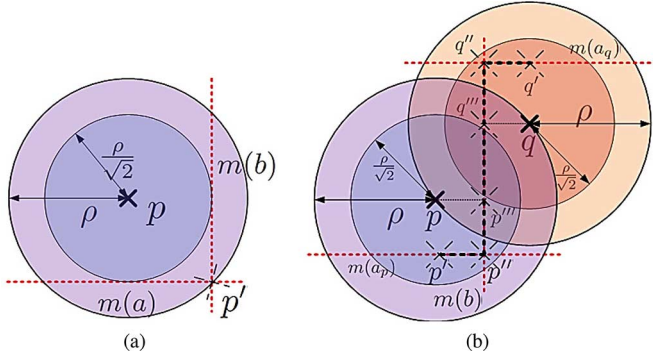


Fig. 6. Two-dimensional sketch: balls and manifolds are presented as circles and lines, respectively. (a) Intersection of two $\rho/\sqrt{2}$ -intersecting manifolds. (b) Construction of a path as defined in Lemma III.3.

decomposed into two perpendicular subspaces, a point $p = (a_1, \dots, a_k, b_1, \dots, b_{d-k}) \in \mathcal{C}$ may be represented as the pair of points (a, b) from subspaces A and B . Under this assumption, the set of manifolds \mathcal{M} consists of two families of k and $(d - k)$ -dimensional manifolds \mathcal{M}^A and \mathcal{M}^B . Family \mathcal{M}^A consists of all manifolds that are defined by fixing a point $a_0 \in A$, while the remaining $d - k$ parameters are variable; \mathcal{M}^B is defined symmetrically. Two manifolds $m(a) \in \mathcal{M}^A$ and $m(b) \in \mathcal{M}^B$ always intersect in exactly one point, i.e., $m(a) \cap m(b) = (a, b) \in \mathcal{C}$. Let $B_r^C(p) = \{q \in \mathcal{C} | \text{dist}(p, q) \leq r\}$ define a ball in \mathcal{C} of radius r centered at $p \in \mathcal{C}$, where dist denotes the Euclidean metric on \mathcal{C} . Likewise, $B_r^B(b)$ and $B_r^A(a)$ denote $(d - k)$ and k -dimensional balls in B and A , respectively.

Definition III.1 (ρ -Intersecting): For $\rho > 0$, we call a manifold $m(a) \in \mathcal{M}^A$ ρ -intersecting for a point $p \in \mathcal{C}$ if $m(a) \cap B_\rho^C(p) \neq \emptyset$, i.e., if $a \in B_\rho^A(p_A)$, where p_A is the projection of p onto A . Similarly for manifolds in B .

A feasible path γ is a continuous mapping from the interval $[0, 1]$ into $\mathcal{C}_{\text{free}}$. The image of a path is defined as $\text{Im}(\gamma) = \{\gamma(\alpha) | \alpha \in [0, 1]\}$. We show that for any collision-free path $\gamma_{p,q}$ of clearance $\rho > 0$ between two configurations p and q the MMS constructs a path from p to q such that: (i) the path lies on the FSCs of the sampled manifolds and (ii) every point on the path is at distance at most ρ from $\gamma_{p,q}$, with a positive probability. Moreover, the probability of failing to find such a path by the MMS algorithm decreases exponentially with the number of samples.

Lemma III.2: For $p \in \mathcal{C}$ and $\rho > 0$ let $m(a) \in \mathcal{M}^A$ and $m(b) \in \mathcal{M}^B$ be two manifolds that are $\rho/\sqrt{2}$ -intersecting for p . Their intersection point $p' = (a, b) = m(a) \cap m(b)$ is in $B_\rho^C(p)$.

For an illustration of Lemma III.2, see Fig. 6(a). The proof (given in [35]) follows immediately from elementary properties of \mathbb{R}^d . Lemma III.3 is the centerpiece of the completeness proof as it establishes connections between two manifolds. Informally, it shows that for any two points p and q , a manifold $m(b) \in \mathcal{M}^B$ that is close to both points enables a connection between two manifolds $m(a_p), m(a_q) \in \mathcal{M}^A$ that are close to p and q , respectively.

Lemma III.3: Let $p, q \in \mathcal{C}$ be two points such that $\text{dist}(p, q) \leq \rho$ and let $m(a_p), m(a_q) \in \mathcal{M}^A$ be two $\rho/\sqrt{2}$ -intersecting manifolds for p and q , respectively. Let

$m(b) \in \mathcal{M}^B$ be a manifold that is simultaneously $\rho/\sqrt{2}$ -intersecting for p and q and let $p' = (a_p, p_B) \in B_\rho^C(p)$ and $q' = (a_q, q_B) \in B_\rho^C(q)$ be the projection of p and q on $m(a_p)$ and $m(a_q)$, respectively.

There exists a path $\gamma_{p',q'}$ between p' and q' such that $\text{Im}(\gamma_{p',q'}) \subseteq (B_\rho^C(p) \cup B_\rho^C(q)) \cap (m(a_p) \cup m(b) \cup m(a_q))$, i.e., there is a path lying on the manifolds within the union of the balls.

Proof: Let $p'' = m(a_p) \cap m(b) = (a_p, b)$ and $q'' = m(a_q) \cap m(b) = (a_q, b)$ denote the intersection point of $m(a_p)$ and $m(a_q)$ with $m(b)$, respectively. Moreover, let $p''' = (p_A, b) \in B_\rho^C(p)$ and $q''' = (q_A, b) \in B_\rho^C(q)$ denote the projections of p and q on $m(b)$. We show that the path which is the concatenation of the segments $(p', p''), (p'', p'''), (p''', q'''), (q''', q'')$ and (q'', q') lies on $(m(a_p) \cup m(b) \cup m(a_q))$ within the union of the balls $(B_\rho^C(p) \cup B_\rho^C(q))$. See Fig. 6(b).

By Lemma III.2 the intersection points p'' and q'' are inside $B_\rho^C(p)$ and $B_\rho^C(q)$, respectively. Thus, by convexity of each ball the segments $(p', p'') \subset m(a_p)$ and $(q', q'') \subset m(a_q)$ as well as the segments $(p'', p'''), (q'', q''') \subset m(b)$ are in $(B_\rho^C(p) \cup B_\rho^C(q))$.

It remains to show that $(p''', q''') \subset m(b)$ is inside $(B_\rho^C(p) \cup B_\rho^C(q))$. Recall that $\text{dist}(p, q) \leq \rho$ and, therefore, $\text{dist}(p''', q''') \leq \rho$. Let \bar{p} be a point on the segment (p''', q''') that, w.l.o.g, is closer to p''' . Thus, $\text{dist}(\bar{p}, p''') \leq \rho/2$. The manifold $m(b)$ is $\rho/\sqrt{2}$ -intersecting, thus $\text{dist}(p, p''') \leq \rho/\sqrt{2}$. As the segments (p, p''') and (p''', \bar{p}) are perpendicular it holds

$$\begin{aligned} \text{dist}(p, \bar{p}) &= \sqrt{\text{dist}(p, p''')^2 + \text{dist}(p''', \bar{p})^2} \\ &\leq \sqrt{\rho^2/2 + \rho^2/4} \\ &< \rho. \end{aligned}$$

Theorem III.4: Let p, q be points in $\mathcal{C}_{\text{free}}$ such that there exists a collision-free path $\gamma_{p,q} \in \Gamma$ of length L and clearance ρ between p and q . Then, the probability of the MMS algorithm to return a path between p and q after generating n_A and n_B manifolds from families \mathcal{M}^A and \mathcal{M}^B as above, respectively is

$$\begin{aligned} \Pr[(p, q)\text{SUCCESS}] &= 1 - \Pr[(p, q)\text{FAILURE}] \\ &\geq 1 - \left\lceil \frac{L}{\rho} \right\rceil [(1 - \mu_A)^{n_A} + (1 - \eta_B)^{n_B}] \end{aligned}$$

where μ_A and η_B are some positive constants smaller than 1.

The constants μ_A and η_B reflect the probability of a manifold to be $\rho/\sqrt{2}$ -intersecting for one or two nearby points, respectively. The proof for Theorem III.4 is rather technical. It involves using Lemma III.3 repeatedly for points along the path $\gamma_{p,q}$ of distance less than ρ . We omit the details and refer the reader to [35] for the full proof.

Recursive Application: The proof of Theorem III.4 assumes that the samples are taken using full high-dimensional manifolds. However, Section II demonstrates a recursive application of MMS where the approximate samples are generated by another application of MMS.

In order to obtain a completeness proof for the two-level scheme let γ be a path of clearance 2ρ . First, assume that the

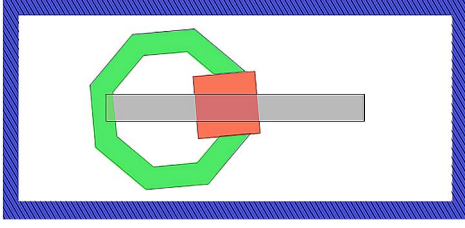


Fig. 7. Tightest solvable Pacman scenario. The gray rectangle shows valid placements of the square such that the Pacman can engulf it without colliding with the scene's bounding box.

samples taken by the first level of MMS are exact. Applying Theorem III.4 for γ and ρ shows that with sufficient probability MMS would find a set M' of manifolds that would contain a path γ' . Since we required clearance 2ρ but relied on the tighter clearance ρ , it is guaranteed that γ' still has clearance ρ . Now, each manifold $m' \in \mathcal{M}'$ is actually only an approximation constructed by another application of MMS. Thus, for each $m' \in \mathcal{M}'$, apply Theorem III.4 to the subpath $\gamma'_{m'} = \gamma' \cap m'$ which has clearance ρ . Concatenation of all the resulting subpaths concludes the argument. Of course the parameters in the inequality in Theorem III.4 change accordingly.

We remark that the recursive approach imposes a mild restriction on the sampling scheme as the sampling and the approximation must be somewhat coordinated. Since in theory $m(a) \cap m(b) = (a, b)$ we must ensure that points that we sample from A are contained in every approximation of $m(b) \in \mathcal{M}^B$ and *vice versa*. In our implementation, this is ensured by restricting the set of possible angles to those used to approximate $m(b) \in \mathcal{M}^B$ (see Section II).

IV. ON THE DIMENSION OF NARROW PASSAGES

Consider the Pacman scenario illustrated in Fig. 3(c) of the experiments section. We obtain a narrow passage by increasing the size of the square-shaped robot making it harder for the Pacman to swallow it. Fig. 4 shows that our approach is significantly less sensitive to this tightening of the free space than the PRM algorithm. In order to explain this, let us take a closer look at the nature of the narrow passage for the tightest solvable case.

In order to get from the start placement to the goal placement, the Pacman must swallow the square, rotate around it and spit it out again. We concentrate on the swallowing motion. Fig. 7 depicts the tightest case, i.e., when the square robot fits exactly into the “mouth” of the Pacman. The gray rectangle indicates the positions of the reference point of the square such that there is a valid movement of the Pacman, considering the walls of the room, that will allow it to swallow the square robot (two-dimensional region, two parameters). The rotation angle of the square is also important (one additional parameter). The range of concurrently possible values for all three parameters is small but does not tend to zero either. The passage becomes only narrow by the fact that the rotation angle of the Pacman must correlate exactly with the orientation of the square to allow for passing through the mouth. Moreover, the set of valid placements for the reference point of the Pacman while swallowing the square (other parameters being fixed) is a line, i.e., its x and y parameter values are coupled. Thus, the passage is a four-dimensional

object as we have a tight coupling of two pairs of parameters in a six-dimensional C-space.

The PRM approach has difficulties to sample in this passage since the measure tends to zero as the size of the square increases. On the other hand, for our approach the passage is only narrow with respect to the correlation of the two angles. As soon as the MMS samples an (approximated) volume that fixes the square robot such that the Pacman *can* engulf it, the approximation of the volume just needs to include a horizontal slice of a suitable angle and the passage becomes evident in the corresponding Minkowski sum computation.

A. Definition of Narrow Passages

Intuition may suggest that narrow passages are tunnel-shaped. However, a one-dimensional tunnel in a high-dimensional C-spaces would correspond to a simultaneous coupling of *all* parameters, which is often not the case. For instance, the discussion of the Pacman scenario shows that the passage is narrow but that it is still a four-dimensional volume, which proved to be a considerable advantage for our approach in the experiments. Although some sampling-based approaches try to take the dimension of a passage into account (see, e.g., [13]) it seems that this aspect is not reflected by existing definitions that attempt to capture attributes of the C-space. Definitions such as ϵ -goodness [24] and expansiveness [21] are able to measure the size of a narrow passage better than the clearance [23] of a path, but neither incorporates the dimension of a narrow passage in a very accessible way. Therefore, we would like to propose a new set of definitions that attempt to simultaneously grasp the narrowness and the dimension of a passage.

We start by defining the “ordinary” clearance of a path in $\mathcal{C}_{\text{free}}$. The characterization is based on the notion of homotopy classes of paths with respect to a set $\Gamma_{s,t}$, i.e., the set of all paths starting at s and ending at t . For a path $\gamma_0 \in \Gamma_{s,t}$ and its homotopy class $\mathcal{H}(\gamma_0)$, we define the clearance of the class as the largest clearance found among all paths in $\mathcal{H}(\gamma_0)$.

Definition IV.1: The **clearance** of a homotopy class $\mathcal{H}(\gamma_0)$ for $\gamma_0 \in \Gamma_{s,t}$ is

$$\sup_{\gamma \in \mathcal{H}(\gamma_0)} \left\{ \sup \{ \rho > 0 \mid B_\rho^d \oplus \text{Im}(\gamma) \subseteq \mathcal{C}_{\text{free}} \} \right\}$$

where \oplus denotes the Minkowski sum of two sets, which is the vector sum of the sets.

By using a d -dimensional ball this definition treats all directions equally, thus considering the passage of $\mathcal{H}(\gamma_0)$ to be a one-dimensional tunnel. We next refine this definition by using a k -dimensional disk, which may be placed in different orientations depending on the position along the path.

Definition IV.2: For some integer $0 < k \leq d$ the **k -clearance** of $\mathcal{H}(\gamma_0)$ is:

$$\sup_{\gamma \in \mathcal{H}(\gamma_0)} \left\{ \rho > 0 \mid \forall t \in [0, 1] \exists \mathbf{R} \in \mathcal{R} : \gamma(t) \oplus \mathbf{R}(t) B_\rho^k \subseteq \mathcal{C}_{\text{free}} \right\}$$

where \mathcal{R} is the set of d -dimensional rotation matrices and B_ρ^k is the k -dimensional ball of radius ρ . In case \mathbf{R} is required to change continuously we talk about **continuous k -clearance**.

Clearly, the k -clearance of $\mathcal{H}(\gamma_0)$ for $k = d$ is simply the clearance of $\mathcal{H}(\gamma_0)$. For decreasing values of k , the k -clearance of a homotopy class is a monotonically increasing sequence. We

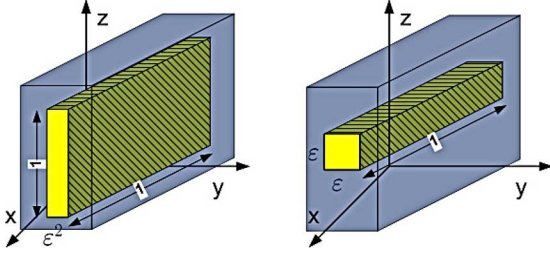


Fig. 8. Two three-dimensional C-spaces consisting of a narrow passage (yellow) surrounded by obstacles (blue).

next define the dimension of a passage using this sequence, that is, we set the dimension to be the first k for which the clearance becomes *significantly larger*⁸ than the original d -dimensional clearance.

Definition IV.3: A passage for $\mathcal{H}(\gamma_0)$ in \mathbb{R}^d of clearance ρ (see Definition IV.1) is called **$d - k + 1$ -dimensional** if k is the largest index such that k -clearance($\mathcal{H}(\gamma_0)$) $\gg \rho$. If for every k , k -clearance($\mathcal{H}(\gamma_0)$) $\not\gg \rho$, then we call the passage one-dimensional.⁹

For instance, both passages in Fig. 8 have a measure of ε^2 thus for a PRM like planner, sampling in either passage is equally hard as the probability of a uniform point sample to lie in either one of the narrow passages is proportional to ε^2 . However, the two passages are fundamentally different. The passage depicted on the right-hand side is a one-dimensional tunnel corresponding to a tight coupling of the three parameters. The passage depicted on the left-hand side is a two-dimensional flume which is much easier to intersect by a probabilistic approach that uses manifolds as samples. Our new definitions formally reveal this difference. For k equals 3, 2 and 1, the k -clearance of the right passage is ε , $\sqrt{2}\varepsilon$ and larger than 1, respectively. For the left passage, this sequence is ε^2 for $k = 3$ and larger than 1 for $k = 2, 1$ which characterizes the passage as two-dimensional.

B. Discussion

We believe that the definitions introduced in Section IV-A, can be an essential component of a formal proof that shows the advantage of manifold samples over point samples in the presence of high-dimensional narrow passages. We sketch the argument briefly. Let $\mathcal{C}_{\text{free}}$ contain a narrow passage of dimension k , that is, the passage has clearance ρ and k -clearance λ , where $\lambda \gg \rho$. This implies that it is possible to place discs of dimension k and radius $\lambda \gg \rho$ into the tight passage. The main argument is that for a random linear manifold of dimension $d - k$ the probability to hit such a disc is proportional to λ , which is much larger than ρ . The probability also depends on the angle between the linear subspace containing the disc and the linear manifold. However, by choosing a proper set of manifold families it is possible to guarantee the existence of at least one family for which this angle is bounded, independent of the orientation of the disc.

⁸We leave this notion informal as it might depend on the problem at hand.

⁹For simplicity of definition, we chose to stop at the largest index k for which k -clearance $\gg \rho$. One could contemplate alternative more elaborate definitions that keep on searching for even larger clearance for smaller indices.

V. FURTHER WORK

The extension of MMS [36] presented here is part of our ongoing efforts towards the goal of creating a general scheme for exploring high-dimensional C-spaces that is less sensitive to narrow passages than currently available tools. As discussed in Section I-C the original scheme imposes a set of conditions that in combination restrict an application of MMS to rather low dimensions. In this paper, we chose to relax condition **C3**, for example, by computing only approximations of three-dimensional manifolds. An alternative path is to relax condition **C2**, for example, by not sampling the manifolds uniformly and independently at random. This would enable the use of manifolds of low dimension as it allows to enforce intersection. Following this path we envision a single-query planner that explores a C-space in an RRT-like fashion. Using these extensions, we wish to apply the scheme to a variety of difficult problems including assembly maintainability (part removal for maintenance [49]) by employing a single-query variant of the scheme.

Another possibility is to explore other ways to compute approximative manifold samples, for instance, the (so far) exact representations of FSCs could be replaced by much simpler (and thus faster) but conservative¹⁰ approximations. This is certainly applicable to manifold samples of dimension one or two and should also enable manifold samples of higher dimensions. We remark that the use of approximations should not harm the probabilistic completeness as long as it is possible to refine the approximations such that they converge to the exact results.

In order to demonstrate the potential of the scheme, we adapted our motion planner to the problem of *three-handed translational* assembly planning. In assembly planning [18], [46], we are given a collection of parts, and the goal is to assemble the parts into one (given) object. Typically, the problem is tackled by starting at the end configuration and recursively separating the object into subgroups. Informally, the number of groups that may be considered *simultaneously* is the number of hands used. The problem, which is known in general to be computationally hard (see, e.g., [22]), has been studied extensively for two hands but little has been done for more.

The first assembly-planning problem we consider, depicted in Fig. 9(a) demonstrates a scenario where the two smoothly-textured parts need to move in alternations in order to exit a surrounding part (the obstacle) in order to reach a disassembled configuration. This problem was solve by our planner within 37 s (average over ten runs) and could not be solved by the PRM algorithm (which was terminated after 10 min). The RRT algorithm managed solving this scenario within 160 s with a success rate of 50% (if the solution was not found within 10 min the run was considered unsuccessful; average over ten runs). The second assembly-planning problem, depicted in Fig. 9(b) demonstrates a scenario where multiple smoothly-textured purple parts need to be moved out of the surrounding green part textured with stripes. At each iteration, two purple parts are chosen at random and the planner attempts to translate both parts (as independent parts) out of the obstacle. The most interesting case occurs when the lower triangles have been removed and the two M-shaped parts need to translate out of the obstacle. In order for this to

¹⁰Approximated FSCs are contained in $\mathcal{C}_{\text{free}}$.

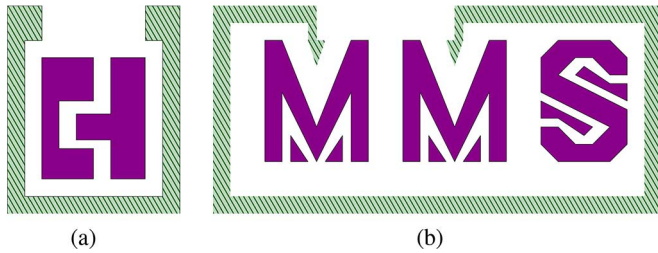


Fig. 9. Three-handed assembly-planning scenarios. Static obstacle (first hand) is depicted in green with a striped texture. Moving parts (second and third hands) are colored with smooth texture. In (b), at each iteration two moving parts are chosen randomly while the remaining parts are considered part of the static obstacles (if they were not disassembled in previous iterations).

occur, the left M-shaped part needs to translate to the bottom-left corner for the right M-shaped part to be able to translate out of the obstacle. Our planner manages to plan this in under 1 s (average over ten runs) while this case could not be solved by either the RRT or PRM algorithms (which were terminated after 10 min). We note that this is not the traditional assembly-planning formulation as the parts are not touching each other. In order for a sampling-based algorithm (such as MMS) to be applicable, some slack is required between the parts. However, the slack can be much smaller when using MMS as opposed to standard sampling-based planners.

Finally, we intend to extend the scheme to and experiment with motion-planning problems for highly redundant robots as well as for fleets of robots, exploiting the symmetries in the respective C-space.

For supplementary material, omitted here for lack of space, the reader is referred to our project web-page <http://acg.cs.tau.ac.il/projects/mms>.

REFERENCES

- [1] N. M. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *Proc. ICRA*, 1996, pp. 113–120.
- [2] B. Aronov and M. Sharir, "On translational motion planning of a convex polyhedron in 3-space," *SIAM J. Comput.*, vol. 26, no. 6, pp. 1785–1803, 1997.
- [3] F. Avnaim, J.-D. Boissonnat, and B. Faverjon, "A practical exact motion planning algorithm for polygonal object amidst polygonal obstacles," in *Proc. Workshop Geom. Robot.*, London, U.K., 1989, pp. 67–86.
- [4] N. Ayanian and V. Kumar, "Decentralized feedback controllers for multiagent teams in environments with obstacles," *IEEE Trans. Robotics*, vol. 26, no. 5, pp. 878–887, Oct. 2010.
- [5] S. Basu, R. Pollack, and M.-F. Roy, "Algorithms in real algebraic geometry," in *Algorithms and Computation in Mathematics*, 2nd ed. Berlin, Germany: Springer-Verlag, 2006.
- [6] E. Berberich, E. Fogel, D. Halperin, M. Kerber, and O. Setter, "Arrangements on parametric surfaces II: Concretizations and applications," *Math. Comput. Sci.*, vol. 4, pp. 67–91, 2010.
- [7] E. Berberich, E. Fogel, D. Halperin, K. Mehlhorn, and R. Wein, "Arrangements on parametric surfaces I: General framework and infrastructure," *Math. Comput. Sci.*, vol. 4, pp. 45–66, 2010.
- [8] D. Berenson, S. S. Srinivasa, and J. J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *I. J. Robot. Res.*, vol. 30, no. 12, pp. 1435–1460, 2011.
- [9] J. F. Canny, *Complexity of Robot Motion Planning (ACM Doctoral Dissertation Award)*. Cambridge, MA, USA: MIT Press, Jun. 1988.
- [10] B. Chazelle, H. Edelsbrunner, L. J. Guibas, and M. Sharir, "A singly exponential stratification scheme for real semi-algebraic varieties and its applications," *Theor. Comput. Sci.*, vol. 84, no. 1, pp. 77–105, 1991.
- [11] H. Choset, W. Burgard, S. Hutchinson, G. Kantor, L. E. Kavraki, K. Lynch, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. Cambridge, MA, USA: MIT Press, Jun. 2005.
- [12] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robot. Autom. Mag.*, vol. 19, no. 4, pp. 72–82, Dec. 2012.
- [13] S. Dalibard and J.-P. Laumond, "Linear dimensionality reduction in random motion planning," *I. J. Robot. Res.*, vol. 30, no. 12, pp. 1461–1476, 2011.
- [14] M. De Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*, 3rd ed. New York, NY, USA: Springer, 2008.
- [15] E. Foegl, D. Halperin, and R. Wein, *CGAL Arrangements and Their Applications*. Berlin, Germany: Springer-Verlag, 2012.
- [16] E. Fogel and D. Halperin, "Exact and efficient construction of Minkowski sums of convex polyhedra with applications," *Comput. Aided Design*, vol. 39, no. 11, pp. 929–940, 2007.
- [17] P. Hachenberger, "Exact minkowski sums of polyhedra and exact and efficient decomposition of polyhedra into convex pieces," *Algorithmica*, vol. 55, no. 2, pp. 329–345, 2009.
- [18] D. Halperin, J.-C. Latombe, and R. H. Wilson, "A general framework for assembly planning: The motion space approach," *Algorithmica*, vol. 26, no. 3–4, pp. 577–601, 2000.
- [19] D. Halperin and M. Sharir, "A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment," *Disc. Comput. Geom.*, vol. 16, no. 2, pp. 121–134, 1996.
- [20] S. Hirsch and D. Halperin, "Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane," in *Proc. WAFR*, 2002, pp. 225–241.
- [21] D. Hsu, J. Latombe, and R. Motwani, "Path planning in expansive configuration spaces," *Int. J. Comp. Geo. App.*, vol. 4, pp. 495–512, 1999.
- [22] L. E. Kavraki and M. N. Kolountzakis, "Partitioning a planar assembly into two connected parts is np-complete," *Inf. Process. Lett.*, vol. 55, no. 3, pp. 159–165, 1995.
- [23] L. E. Kavraki, M. N. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. Robot. Autom.*, vol. 14, no. 1, pp. 166–171, Feb. 1998.
- [24] L. E. Kavraki, J.-C. Latombe, R. Motwani, and P. Raghavan, "Randomized query processing in robot path planning," *J. Comput. Syst. Sci.*, vol. 57, no. 1, pp. 50–60, Aug. 1998.
- [25] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Trans. Robot. Autom.*, vol. 12, no. 4, pp. 566–580, Aug. 1996.
- [26] J. J. Kuffner, "Effective sampling and distance metrics for 3D rigid body path planning," in *Proc. ICRA*, 2004, pp. 3993–3998.
- [27] J. J. Kuffner and S. M. Lavalle, "RRT-Connect: An efficient approach to single-query path planning," in *Proc. ICRA*, 2000, pp. 995–1001.
- [28] A. M. Ladd and L. E. Kavraki, "Generalizing the analysis of PRM," in *Proc. ICRA*, May 2002, pp. 2120–2125.
- [29] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer, 1991.
- [30] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep. 98-11, 1998.
- [31] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [32] J.-M. Lien, "Hybrid motion planning using Minkowski sums," in *Proc. RSS*, 2008, pp. 97–104.
- [33] T. Lozano-Perez, "Spatial planning: A configuration space approach," MIT AI Memo 605, 1980.
- [34] J. H. Reif, "Complexity of the mover's problem and generalizations," in *Proc. FOCS*, Washington, DC, USA, 1979, pp. 421–427, IEEE Comput. Soc.
- [35] O. Salzman, M. Hemmer, and D. Halperin, "On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages," *CoRR*, 2012, abs/1202.5249.
- [36] O. Salzman, M. Hemmer, B. Raveh, and D. Halperin, "Motion planning via manifold samples," *Algorithmica*, vol. 67, no. 4, pp. 547–565, 2013.
- [37] J. T. Schwartz and M. Sharir, "On the 'piano movers' problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers," *Commun. Pure Appl. Math.*, vol. 35, pp. 345–398, 1983.

- [38] J. T. Schwartz and M. Sharir, "On the "piano movers" problem: II. General techniques for computing topological properties of real algebraic manifolds," *Adv. Appl. Math.*, vol. 4, no. 3, pp. 298–351, 1983.
- [39] M. Sharir, *Algorithmic Motion Planning, Handbook of Discrete and Computational Geometry*, 2nd ed. Boca Raton, FL, USA: CRC Press, 2004.
- [40] J. G. Siek, L.-Q. Lee, and A. Lumsdaine, *The Boost Graph Library: User Guide and Reference Manual*. Reading, MA, USA: Addison-Wesley, 2001.
- [41] K. Solovey, O. Salzman, and D. Halperin, "Finding a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning," in *Proc. WAFR*, 2014, to appear.
- [42] M. Stilman, "Task constrained motion planning in robot joint space," in *Proc. IROS*, 2007, pp. 3074–3081.
- [43] The CGAL Project, CGAL User and Reference Manual, 2010. [Online]. Available: <http://www.cgal.org/>, CGAL Editorial Board, 3.7 edition
- [44] G. Wagner, M. Kang, and H. Choset, "Probabilistic path planning for multiple robots with subdimensional expansion," in *Proc. ICRA*, 2012, pp. 2886–2892.
- [45] R. Wein, "Exact and efficient construction of planar Minkowski sums using the convolution method," *ESA*, pp. 829–840, 2006.
- [46] R. H. Wilson and J.-C. Latombe, "Geometric reasoning about mechanical assembly," *Artif. Intelligence*, vol. 71, pp. 371–396, 1994.
- [47] J. Yang and E. Sacks, "RRT path planner with 3 DOF local planner," in *Proc. ICRA*, 2006, pp. 145–149.
- [48] H.-Y. Yeh, S. L. Thomas, D. Eppstein, and N. M. Amato, "UOBPRM: A uniformly distributed obstacle-based PRM," in *Proc. IROS*, 2012, pp. 2655–2662.
- [49] L. Zhang, X. Huang, Y. J. Kim, and D. Manocha, "D-plan: Efficient collision-free path computation for part removal and disassembly," *J. Comput. Aided Design Appl.*, vol. 5, pp. 774–786, 2008.



Oren Salzman is currently working towards the Ph.D. degree at the School for Computer Science, Tel-Aviv University, Tel Aviv, Israel.



Michael Hemmer was a Postdoctoral Fellow at the School for Computer Science, Tel-Aviv University, Tel Aviv, Israel, during the time of this study and is now a Researcher with the Institute of Operating Systems and Computer Networks, University of Technology Braunschweig, Braunschweig, Germany.



Dan Halperin is a Professor with the School for Computer Science, Tel-Aviv University, Tel Aviv, Israel.