Algorithmica (2013) 67:547–565
DOI 10.1007/s00453-012-9736-1

Motion Planning via Manifold Samples

Oren Salzman · Michael Hemmer · Barak Raveh · Dan Halperin

Received: 31 January 2012 / Accepted: 17 December 2012 / Published online: 9 January 2013
© Springer Science+Business Media New York 2013

Abstract We present a general and modular algorithmic framework for path planning of robots. Our framework combines geometric methods for exact and complete analysis of low-dimensional configuration spaces, together with practical, considerably simpler sampling-based approaches that are appropriate for higher dimensions. In order to facilitate the transfer of advanced geometric algorithms into practical use, we suggest taking samples that are *entire low-dimensional manifolds of the configuration space* that capture the connectivity of the configuration space much better than isolated point samples. Geometric algorithms for analysis of low-dimensional manifolds then provide powerful primitive operations. The modular design of the framework enables independent optimization of each modular component. Indeed, we have developed, implemented and optimized a primitive operation for complete and exact combinatorial analysis of a certain set of manifolds, using arrangements of curves of rational functions and concepts of generic programming. This in turn enabled us to implement our framework for the concrete case of a polygonal robot translating

This work has been supported in part by the 7th Framework Programme for Research of the European Commission, under FET-Open grant number 255827 (CGL—Computational Geometry Learning), by the Israel Science Foundation (grant no. 1102/11), by the German-Israeli Foundation (grant no. 969/07), and by the Hermann Minkowski–Minerva Center for Geometry at Tel Aviv University.

O. Salzman (✉) · M. Hemmer · B. Raveh · D. Halperin
Tel-Aviv University, P.O.B. 39040, Tel-Aviv 69978, Israel
e-mail: orenzalz@post.tau.ac.il

M. Hemmer
e-mail: mhsaar@googlemail.com

B. Raveh
e-mail: barak.raveh@mail.huji.ac.il

D. Halperin
e-mail: danha@post.tau.ac.il

B. Raveh
The Hebrew University, The Edmond J. Safra Campus, Givat Ram, Jerusalem 91904, Israel

and rotating amidst polygonal obstacles. We show that this instance of the framework is probabilistically complete. Moreover, we demonstrate that the integration of several carefully engineered components leads to significant speedup over the popular PRM sampling-based algorithm, which represents the more simplistic approach that is prevalent in practice.

Keywords Motion planning · Computational geometry · Manifolds

1 Introduction

Motion planning is a fundamental research topic in robotics with applications 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 [11, 25, 27]. In its basic form, the motion-planning problem is to find a collision-free path for a robot or a moving object R in a *workspace* cluttered with static obstacles. The spatial pose of R, or the *configuration* of R, is uniquely defined by some set of parameters, the degrees of freedom (*dof*s) of R. The set of all robot configurations $\mathcal{C}$ is termed the *configuration space* of the robot, and decomposes into the disjoint sets of free and forbidden configurations, namely $\mathcal{C}_{\text{free}}$ and $\mathcal{C}_{\text{forb}}$, respectively. Thus, it is common to rephrase the motion-planning problem as the problem of moving R from a start configuration q_s to a target configuration q_t in a path that is fully contained within $\mathcal{C}_{\text{free}}$.

Analytic Solutions to the General Motion Planning Problem The motion-planning problem is computationally hard with respect to the number of *dof*s [33], yet much research has been devoted to solving the general problem and its various instances using geometric, algebraic and combinatorial tools. The configuration-space formalism was introduced by Lozano-Perez [30] in the early 1980s. Schwartz and Sharir proposed the first general algorithm for solving the motion planning problem, with running time that is doubly-exponential in the number of *dof*s [37]. Singly exponential-time algorithms have followed [4, 9, 10], but are generally considered too complicated to be implemented in practice.

Solutions to Low-Dimensional Instances of the Problem Although the general motion-planning problem cannot be efficiently solved analytically, more efficient algorithms have been proposed for various low-dimensional instances [25], such as translating a polygonal or polyhedral robot [2, 30], and translation with rotation of a polygonal robot in the plane [3, 18, 36]. For a survey of related approaches see [38]. In recent years, considerable advances in robust implementation of computational geometry algorithms have led to a set of implemented tools that are of interest in this context. Minkowski sums, which allow representation of the configuration space of a translating robot, have robust and exact planar and three-dimensional implementations [15, 17, 42]. Likewise, implementations of planar arrangements[1] for curves [16, 40], are essential components in [37].

[1] A subdivision of the plane into zero-dimensional, one-dimensional and two-dimensional cells, called vertices, edges and faces, respectively induced by the curves.

Sampling-Based Approaches to Motion Planning The sampling-based approach to motion-planning has extended the applicability of motion planning algorithms beyond the restricted subset of problems that can be solved efficiently by exact algorithms [11, 27]. Sampling-based motion planning algorithms, such as Probabilistic Roadmaps (PRM) [22], Expansive Space Trees (EST) [20] and Rapidly-exploring Random Trees (RRT) [26], 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 robot configurations. This can be done either in a multi-query setting, to efficiently answer multiple queries for the same scenario, as in the PRM algorithm, or in a single-query setting, as in the RRT and EST algorithms. For a general survey on the subject see [11]. Importantly, the PRM and RRT algorithms were both shown to be probabilistically complete [21, 23, 24], 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 run-time, and the proper usage of sampling-based approaches may still be considered somewhat of an art. Moreover, sampling-based methods are also considered sensitive to narrow passages in the configuration space, due to the high-probability of missing the passage.

This work is not the first to consider manifolds in the context of sampling-based motion planning. Several works considered manifolds representing configurations that adhere to some constraint on the robot (e.g., where torque constraints are valid). Berenson et al. [7] presented a planner that projects random samples onto such manifolds while Porta and Jaillat [32] presented a planner that plans on these constrained manifolds using numerical continuation methods for high dimensions, which provide numerical procedures to describe an implicitly defined variety using a set of local charts.

Hybrid Methods for Motion-Planning Few hybrid methods attempt to combine both deterministic and probabilistic planning strategies. Hirsch and Halperin [19] studied two-disc motion planning by exactly decomposing the configuration space of each robot, then combining the two solutions to a set of free, forbidden and mixed cells, and using PRM to construct the final connectivity graph. Zhang et al. [44] used PRM in conjunction with approximate cell decomposition, which also divides space to free, forbidden and mixed cells. Other studies have suggested to connect a dense set of near-by configuration space "slices". Each slice is decomposed to free and forbidden cells, but adjacent slices are connected in an inexact manner, by e.g., identifying overlaps between adjacent slices [13, pp. 283–287], or heuristic interpolation and local-planning [28]. In [43] a 6 *dof* RRT planner is presented with a 3 *dof* local planner hybridizing probabilistic, heuristic and deterministic methods.

1.1 Contribution

In this study, we present a novel general scheme for motion planing via manifold samples (MMS), which extends sampling-based techniques like PRM as follows: Instead of sampling isolated robot configurations, we sample *entire low-dimensional manifolds*, which can be analyzed by complete and exact methods for decomposing space. This yields an explicit representation of maximal connected patches of

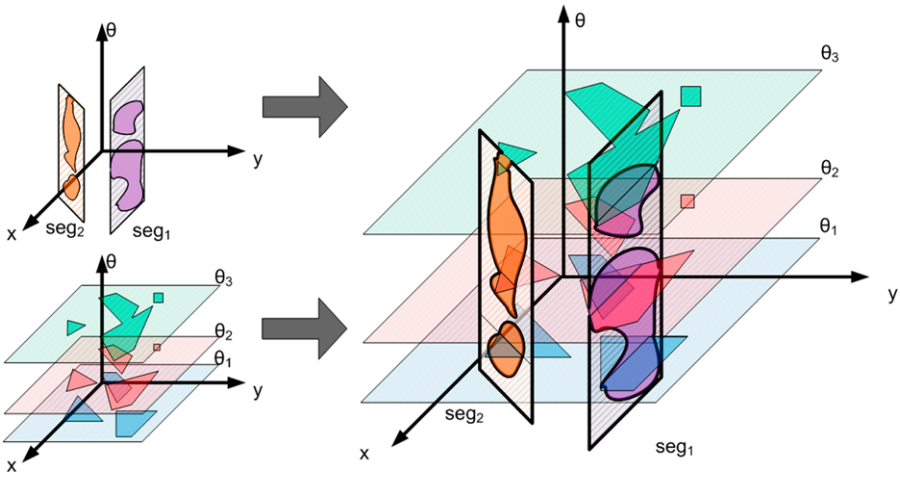


Fig. 1 Three-dimensional configuration space: The *left side* illustrates two families of manifolds where the free cells are *darkly shaded*. The *right side* illustrates their intersection that induces the graph $\mathcal{G}$

free configurations on each manifold, and provides a much better coverage of the configuration space compared to isolated point samples. The manifold samples are deliberately chosen such that they are likely to intersect each other. This allows to establish connections between different manifolds. The general scheme of MMS is illustrated in Fig. 1. A detailed discussion of the scheme is presented in Sect. 2.

In Sect. 3, we discuss the application of MMS to the concrete case of a polygonal robot translating and rotating in the plane amidst polygonal obstacles. We present in detail appropriate families of manifolds as well as filtering schemes that should also be of interest for other scenarios. Section 4 provides a proof of probabilistic completeness for this case. Although our software is prototypical, we emphasize that the achieved results are due to careful design and implementation on all levels. In Sect. 5 we present implementation details and experimental results, which show our method's superior behavior for several test cases vis-à-vis a common implementation of the sampling-based PRM algorithm. For example, in a narrow passage scenario we demonstrate a 58-fold improvement.

In the time that has passed since the submission of this paper, we made considerable progress in developing MMS, which is summarized in [34] and will appear in the upcoming proceedings of WAFR12; for more details see Sect. 6.

2 General Scheme for Planning with Manifold Samples

Preprocessing—Constructing Connectivity Graph We propose a multi-query planner for motion planning problems in a possibly high-dimensional configuration space. The preprocessing stage constructs the *connectivity graph* of $\mathcal{C}$, a data structure that captures the connectivity of $\mathcal{C}$ using manifolds as samples. The manifolds are decomposed into cells in $\mathcal{C}_{\text{free}}$ and $\mathcal{C}_{\text{forb}}$ in a complete and exact manner; we call a cell of the

decomposed manifold that lies in $\mathcal{C}_{\text{free}}$ a *free space cell* (FSC) and refer to the connectivity graph as $\mathcal{G}$. The FSCs serve as nodes in $\mathcal{G}$ while two nodes in $\mathcal{G}$ are connected by an edge if their corresponding FSCs intersect. See Fig. 1 for an illustration.

We formalize the preprocessing stage by considering manifolds induced by a family of constraints Ψ, such that $\psi \in \Psi$ defines a manifold m_ψ of the configuration space. The construction of a manifold m_ψ and its decomposition into FSCs are carried out via a Ψ-primitive (denoted P_Ψ) applied to an element $\psi \in \Psi$. By a slight abuse of notation we refer to an FSC both as a cell and a node in the graph. Using this notation, Algorithm 1 summarizes the construction of $\mathcal{G}$. In lines 3–4, a new manifold constraint is generated and added to the collection of manifold constraints X. In lines 5–7, the manifold induced by the new constraint is decomposed by the appropriate primitive and its FSCs are added to $\mathcal{G}$.

Algorithm 1 Construct Connectivity Graph

1: $V \leftarrow \emptyset$, $E \leftarrow \emptyset$, $X \leftarrow \emptyset$
2: **repeat**
3: $\quad \psi \leftarrow \text{generate\_constraint}(V, E, X)$
4: $\quad X \leftarrow X \cup \{\psi\}$
5: $\quad FSC_{m_\psi} \leftarrow P_\Psi(m_\psi)$
6: $\quad V \leftarrow V \cup \{\text{fsc} \mid \text{fsc} \in FSC_{m_\psi}\}$
7: $\quad E \leftarrow E \cup \{(\text{fsc}_1, \text{fsc}_2) \mid \text{fsc}_1 \in V, \text{fsc}_2 \in FSC_{m_\psi}, \text{fsc}_1 \cap \text{fsc}_2 \neq \emptyset \ \&\ \text{fsc}_1 \neq \text{fsc}_2\}$
8: **until** stopping_condition
9: **return** $G(V, E)$

Query Once the connectivity graph $\mathcal{G}$ has been constructed it can be queried for paths between two configurations q_s and q_t in the following manner: A manifold that contains q_s (respectively q_t) in one of its FSCs is generated and decomposed. Its FSCs and their appropriate edges are added to $\mathcal{G}$. We compute a path p in $\mathcal{G}$ between the FSCs that contain q_s and q_t. A path in $\mathcal{C}_{\text{free}}$ may then be computed by planning a path within each FSC in p.

2.1 Desirable Properties of Manifold Families

Choosing the specific set of manifold families may depend on the concrete problem at hand, as detailed in the next section. However, it seems desirable to retain some general properties: (i) each manifold should be *simple* enough such that it is possible to decompose it into free and forbidden cells in a computationally efficient manner; (ii) the choice of manifold families should *cover* the configuration space, such that each configuration intersects at least a single manifold m_ψ; (iii) local transitions between close-by configurations should be made possible via cross-connections of several intersecting manifolds, which we term the *spanning* property. These properties, defined formally for the concrete case of a polygonal robot translating and rotating in the plane as discussed in Sect. 3 are used to provide a probabilistic completeness proof for that instance in Sect. 4.

2.2 Exploration and Connection Strategies

A naïve way to generate constraints that induce manifolds is by random sampling. Primitives may be computationally complex and should thus be applied sparingly. We suggest a general exploration/connection scheme and additional optimization heuristics that may be used in concrete implementations of the proposed general scheme. We describe strategies in general terms, providing conceptual guidelines for concrete implementations, as demonstrated in Sect. 3.

Exploration and Connection Phases Generation of constraints is done in two phases: *exploration* and *connection*. In the exploration phase constraints are generated such that primitives will produce FSCs that introduce new connected components in $\mathcal{C}_{\text{free}}$. The aim of the exploration phase is to efficiently increase the coverage of the configuration space as much as possible. In contrast, in the connection phase constraints are generated such that primitives will produce FSCs that connect existing connected components in $\mathcal{G}$. Once a constraint is generated, $\mathcal{G}$ is updated as described above. Finally, we note that we can alternate between exploration and connection, namely we can decide to further explore after some connection work has been performed.

Region of Interest (RoI) Decomposing an entire manifold m_ψ by a primitive P_Ψ may be unnecessary. Patches of m_ψ may intersect $\mathcal{C}_{\text{free}}$ in highly explored parts or connect already well-connected parts of $\mathcal{G}$ while others may intersect $\mathcal{C}_{\text{free}}$ in sparsely explored areas or less well-connected parts of $\mathcal{G}$. Identifying the regions where the manifold is of good use (depending on the phase) and constructing m_ψ only in those regions increases the effectiveness of P_Ψ while desirably biasing the samples. We refer to a manifold patch that is relevant in a specific phase as the *Region of Interest*—RoI of the manifold.

Constraint Filtering The goal of the *connection* phase is to improve the connectivity of the graph $\mathcal{G}$. Thus, we would like to use manifolds that are likely to connect FSCs (the nodes of $\mathcal{G}$) that do not yet belong to the same connected component of $\mathcal{G}$. As a consequence we suggest to apply a filter immediately after generating a constraint ψ. That is, we check whether the associated manifold m_ψ could contribute to the connectivity of $\mathcal{G}$. If this not the case we discard the constraint ψ, otherwise we indeed compute the (costly) decomposition of m_ψ into FSCs and update the graph $\mathcal{G}$ accordingly.

3 The Case of Rigid Polygonal Motion

We demonstrate the scheme suggested in Sect. 2 by considering a polygonal robot R translating and rotating in the plane amidst polygonal obstacles. A configuration of R describes the position of the reference point (center of mass) of R and the orientation of R. As we consider full rotations, the configuration space $\mathcal{C}$ is the three dimensional space $\mathbb{R}^2 \times S^1$.

3.1 Manifold Families

As defined in Sect. 2, we consider manifolds defined by *constraints* and construct and decompose them using *primitives*. We suggest the following constraints restricting motions of R and describe their associated primitives: The *Angle Constraint* fixes the orientation of R while it is still free to translate anywhere within the workspace; the *Segment Constraint* restricts the position of the reference point to a segment in the workspace while R is free to rotate.

The left part of Fig. 1 demonstrates decomposed manifolds associated to the angle (left bottom) and segment (left top) constraints. The angle constraint induces a two-dimensional horizontal plane where the cells are polygons. The segment constraint induces a two-dimensional vertical slab where the cells are defined by the intersection of rational curves (as explained in Sect. 5.1).

We delay the discussion of creating and decomposing manifolds to Sect. 5.1. For now, notice that the Segment-Primitive is far more time-consuming than the Angle-Primitive.

3.2 Exploration and Connection Strategies

We use manifolds constructed by the Angle-Primitive for the exploration phase and manifolds constructed by the Segment-Primitive for the connection phase. Since the Segment-Primitive is far more costly than the Angle-Primitive, we focused our efforts on optimizing the former.

Region of Interest—RoI As suggested in Sect. 2.2 we may consider the Segment-Primitive in a subset of the range of angles. This results in a somewhat "weaker" yet more efficient primitive than considering the whole range. If the connectivity of a local area of the configuration space is desired, then using this optimization may suffice while considerably speeding up the algorithm.

Generating Segments Consecutive layers (manifolds of the Angle Constraint) have a similar structure unless topological criticalities occur in $\mathcal{C}$. Once a topological criticality occurs, an FSC either appears and grows or shrinks and disappears. We thus suggest a heuristic for generating a segment in the workspace for the Segment-Primitive using the size of the cell as a parameter where we refer to small and large cells according to pre-defined constants. The RoI used will be proportional to the size of the FSC. The segment generated will be chosen with one of the following procedures which are used in Algorithm 2.

Random Procedure Return a random segment from the workspace.

Large Cell Procedure Return a random segment in the cell.

Small Cell Procedure Intersect the FSC with the next (or the previous) layer. Return a segment connecting a random point from the FSC and a random point in the intersection.

Algorithm 2 Generate Segment Constraint (V, E)

```
if random_num([0, 1]) ≥ random_threshold then
    return random_segment_procedure()
else
    fsc ← random_fsc(V)
    α ← [size(fsc) − small_cell_size]/[large_cell_size − small_cell_size]
    if random_num([0, 1]) ≥ α then
        return small_cell_procedure(fsc, V)
    else
        return large_cell_procedure(fsc, V)
    end if
end if
```

Algorithm 3 Filter Segment (s, RoI, V, E)

```
cc_ids ← ∅
for all v ∈ V do
    fsc ← free_space_cell(v)
    if constraining_angle(fsc) ∈ RoI then
        cc_ids ← cc_ids ∪ connected_component_id(v)
    end if
end for
if |cc_ids| ≤ 1 then
    return filter_out
end if
```

Constraint Filtering As suggested in Sect. 2.2, we avoid computing unnecessary primitives. All FSCs that will intersect a "candidate" constraint s, namely all FSCs of layers in its RoI, are tested. If they are all in the same connected component in $\mathcal{G}$, s can be discarded as demonstrated in Algorithm 3.

3.3 Path Planning Query

For a query $q = (q_s, q_t)$, where $q_s = (x_s, y_s, \theta_s)$ and $q_t = (x_t, y_t, \theta_t)$, $P_\Theta(\theta_s)$ and $P_\Theta(\theta_t)$ are constructed and the FSCs are added to $\mathcal{G}$. A path of FSCs between the FSCs containing q_s and q_t is searched for. A local path in an Angle-Primitive's FSC (which is a polygon) is constructed by computing the shortest path on the visibility graph defined by the vertices of the polygon. A local path in an FSC of a Segment-Primitive (which is an arrangement cell) is constructed by applying cell decomposition and computing the shortest path on the graph induced by the decomposed cells.

4 Probabilistic Completeness for MMS Applied to Rigid Polygonal Motion

We analyze the application of MMS to the case of a polygonal robot rotating and translating in the plane. We show that for any collision-free path $\gamma_{a,b}$ of clearance

$\rho > 0$ between two configurations a and b, it is possible to construct a path from a to b with distance at most ρ from $\gamma_{a,b}$ on the union of manifolds used. Moreover the probability of finding such a path by the MMS algorithm increases exponentially with the number of samples.

Setting A configuration is defined by a three-dimensional point $p = (x_p, y_p, \theta_p)$ where x_p, y_p is the position of the reference point of the robot R in the workspace and θ_p is the amount of counterclockwise rotation of R relative to its original placement. The configuration space is $\mathbb{R}^2 \times S^1$. We assume that the workspace $\mathcal{W}$ is bounded by a box and by scaling both the translational and rotational parts, we consider the configuration space to be $\mathcal{C} = [0, 1]^2 \times [0, 1)$. As $\mathbb{R}^2 \times S^1$ induces a periodic workspace and not a cube, when we consider angle differences, it is done modulo one.

MMS Algorithm We consider two families (types) of manifolds, *horizontal layers* and *vertical lines*.[2] The MMS algorithm samples n_p constraints for the lines, these constraints are points sampled uniformly from the xy-plane $[0, 1]^2$. The MMS algorithm samples n_θ constraints for the layers, these constraints are angles sampled uniformly from the θ-axis $[0, 1)$. For a query (a, b) where $a = (x_a, y_a, \theta_a)$ and $b = (x_b, y_b, \theta_b)$, we assume that the MMS algorithm adds the horizontal layers at angles θ_a, θ_b.

4.1 Notation

We introduce the following notation and definitions used in the proof:

- $dist(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + \min((p_\theta - q_\theta)^2, (2 - p_\theta - q_\theta)^2)}$. This is the periodic Euclidean distance on $\mathcal{C} = [0, 1]^2 \times [0, 1)$ where $\theta = 1$ and $\theta = 0$ are identical.
- $B_r(p) = \{q \in \mathcal{C} \mid dist(p, q) \leq r\}$—a ball in $\mathcal{C}$ of radius r centered at the three-dimensional point p.
- $D_r(\tilde{p}) = \{\tilde{p} \in \mathbb{R}^2 \mid |\tilde{q} - \tilde{p}| \leq r\}$—a disk in $\mathbb{R}^2$ of radius r centered at $\tilde{p} \in [0, 1]^2$.
- $\gamma \in [0, 1] \to \mathcal{C}_{\text{free}}$—a free path γ, i.e., a continuous mapping from the interval $[0, 1]$ into $\mathcal{C}_{\text{free}}$.
- $\Gamma_{a,b} = \{\gamma \in [0, 1] \to \mathcal{C}_{\text{free}} \mid \text{with } \gamma(0) = a \text{ and } \gamma(1) = b\}$—the set off all free paths from a to b.
- $\text{Im}(\gamma) = \{c \in \mathcal{C} \mid \exists \alpha \in [0, 1], \gamma(\alpha) = c\}$ is the image of a path γ.
- $m'_\theta = \{(x, y, \theta') \in \mathcal{C} \mid x, y \in [0, 1]\}$ is a horizontal layer at a fixed angle $\theta' \in [0, 1)$.
- $m_{(x',y')} = \{(x', y', \theta) \in \mathcal{C} \mid \theta \in [0, 1)\}$ is a vertical line at a fixed point $(x', y') \in [0, 1]^2$.
- $M_\Theta = \{m_\theta \mid \theta \in [0, 1)\}$ is the family of horizontal layers.
- $M_P = \{m_{x,y} \mid (x, y) \in [0, 1]^2\}$ is the family of vertical lines.

[2] Considering vertical lines and not vertical slabs as discussed in Sect. 3 is done for simplifying the proof. This is obviously a special case of the same family of manifolds thus the presented proof applies for the case of vertical slabs.

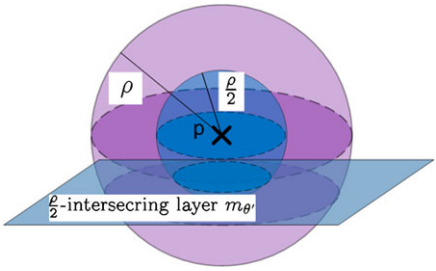


Fig. 2 A $\frac{\rho}{2}$-intersecting layer of a point p

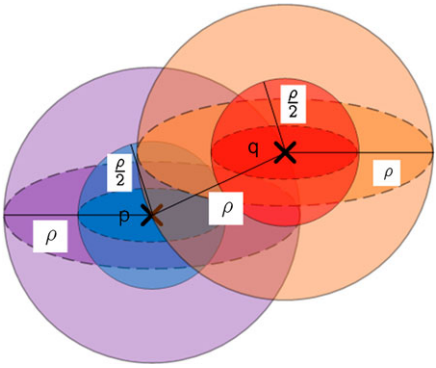


Fig. 3 Points of distance ρ apart inscribed by balls of radii ρ and $\frac{\rho}{2}$

- $\frac{\rho}{2}$-intersecting—a horizontal layer $m_{\theta'}$ is $\frac{\rho}{2}$-intersecting for a point $p \in \mathcal{C}$ and a bound ρ if $m_{\theta'} \cap B_{\frac{\rho}{2}}(p) \neq \emptyset$. See Fig. 2 for an illustration.
- ρ-connecting—a vertical line m_v is ρ-connecting for a point p in $\mathcal{C}$ if the projection of p onto the xy-plane is of distance not greater than $r = \frac{\sqrt{3}}{2}\rho$ from v.
- Pr[(a, b)SUCCESS]—Probability of MMS algorithm to return a path from a to b when such a path exists.
- Pr[(a, b)FAILURE]—Probability of MMS algorithm not to return a path from a to b when such a path exists.

4.2 Probabilistic Completeness Proof

Motivation and Outline In order to construct a path from a source configuration to a target configuration, we first show that for any two close-by configurations, it is possible to construct a path on the manifolds used by MMS such that the path starts from the vicinity of the source configuration and terminates at the vicinity of the target configuration. Two close-by configurations will be formally described as two points with distance no greater than ρ, while the vicinity of a point p will be formally described as $B_{\frac{\rho}{2}}(p)$, the points of distance smaller or equal than $\frac{\rho}{2}$ from p. Manifolds that may induce a path between two close-by configurations are a ($\frac{\rho}{2}$-intersecting) horizontal layer intersecting the vicinity of the first point, a ($\frac{\rho}{2}$-intersecting) horizontal layer intersecting the vicinity of the second point and a (ρ-connecting) vertical line intersecting both layers. Moreover, the properties of the ρ-connecting vertical line will ensure that the path induced by these manifolds is within distance of at most

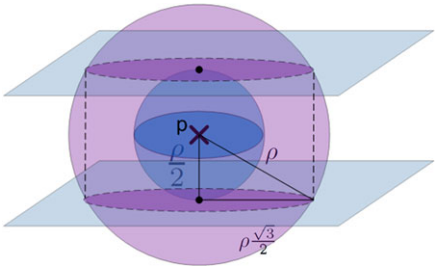


Fig. 4 Minimal area of intersection between a $\frac{\rho}{2}$-intersecting layer and a vertical line within $B_\rho(p)$

ρ of either points. These properties will be shown in Lemmas 1, 2 and 3. Using these lemmas, Theorem 1 will show the probabilistic completeness of the MMS algorithm in this case.

In the following we consider points p, q such that $dist(p, q) \leq \rho$. For each point we consider the balls centered at the points with radii ρ and $\frac{\rho}{2}$. See Fig. 3 for an illustration.

Lemma 1 *Let p be a point in $\mathbb{R}^2 \times S^1$, $m_{\theta'} \in M_\Theta$ be a $\frac{\rho}{2}$-intersecting layer of p and $m_v \in M_P$ be a ρ-connecting vertical line of p. The intersection of the two manifolds is within distance ρ of p, i.e., $p'_v = m_{\theta'} \cap m_v \in B_\rho(p)$.*

Proof Let p' be the vertical projection of $p = (x_p, y_p, \theta_p)$ on $m_{\theta'}$. Since $m_{\theta'}$ is $\frac{\rho}{2}$-intersecting we know that $dist(p, p') \leq \frac{\rho}{2}$. Since m_v is ρ-connecting with p we know that $dist(p', p'_v) \leq \rho\sqrt{3}/2$. Using the Pythagorean Theorem we can conclude

$$dist\big(p, p'_v\big) = \sqrt{dist\big(p, p'\big)^2 + dist\big(p', p'_v\big)} \leq \sqrt{\left(\frac{\rho}{2}\right)^2 + \left(\frac{\sqrt{3}}{2}\rho\right)^2} = \rho.$$

For an illustration see Fig. 4. □

Lemma 2 *Let p and q be two points in $\mathcal{C}$ such that $dist(p, q) \leq \rho$ and let $\tilde{p}$ and $\tilde{q}$ denote their projection onto the xy-plane. Let $m_v \in M_P$ be a vertical line induced by a two-dimensional point v on the xy-plane.*

If $v \in D_{\frac{\sqrt{3}-1}{2}}(\frac{\tilde{p}+\tilde{q}}{2})$ then m_v is ρ-connecting for both p and q.

Proof Let v be a point on the xy-plane inside the disk of radius $\frac{\sqrt{3}-1}{2}\rho$ located midway between $\tilde{p}$ and $\tilde{q}$ (Fig. 5). By the triangle inequality

$$dist(v, \tilde{p}) \leq dist\left(v, \frac{\tilde{p}+\tilde{q}}{2}\right) + dist\left(\frac{\tilde{p}+\tilde{q}}{2}, \tilde{p}\right) \leq \frac{\sqrt{3}-1}{2}\rho + \frac{\rho}{2} = \frac{\sqrt{3}}{2}\rho.$$

Hence v is ρ-intersecting for p. Symmetrically, v is ρ-intersecting for q. □

Lemma 3 *Let p, q be two points such that $dist(p, q) \leq \rho$, $m_{\theta'_p}$, $m_{\theta'_q}$ be two $\frac{\rho}{2}$-intersecting layers of p and q respectively. Let m_v be a ρ-connecting vertical line*

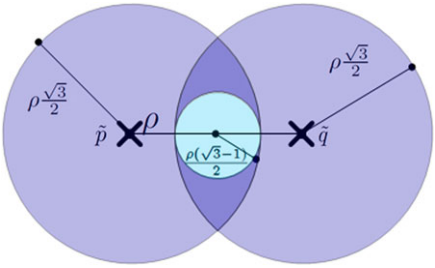


Fig. 5 Projection $\tilde{p}, \tilde{q}$ of two points p and q of distance ρ. *Center disk* is $D_{\frac{\sqrt{3}-1}{2}}(\frac{\tilde{p}+\tilde{q}}{2})$

of both p and q as defined in Lemma 2. Let p' be the projection of p onto $m_{\theta'_p}$, q' be the projection of q onto $m_{\theta'_q}$.

There exists a path $\gamma_{p',q'}$ between p' and q' such that $\mathrm{Im}(\gamma_{p',q'}) \subseteq (m_{\theta'_p} \cup m_v \cup m_{\theta'_q}) \cap (B_\rho(p) \cup B_\rho(q))$.

Proof We construct a path $\gamma_{p',q'}$ by concatenating five paths γ_1, γ_2, γ_3, γ_4, γ_4 as illustrated in Fig. 6 where:

- γ_1 is the segment connecting p' and p'_v, the intersection of $m_{\theta'_p}$ and m_v
- γ_2 is the segment connecting p'_v and p_v, the projection of p onto the m_v
- γ_3 is the segment connecting p_v and q_v, the projection of q onto the m_v
- γ_4 is the path connecting q_v and q'_v, the intersection of m_v and $m_{\theta'_q}$
- γ_5 is the segment connecting q'_v and q'.

As discussed in Lemma 1 the points p', p'_v and p_v are within $B_\rho(p)$. Thus γ_1 and γ_2 are contained in $B_\rho(p)$ due to convexity of $B_\rho(p)$. A symmetric argument applies for γ_4 and γ_5 with respect to $B_\rho(q)$.

Thus we are left to show that $\gamma_3 \subseteq B_\rho(p) \cup B_\rho(q)$: Recall that $dist(p,q) \leq \rho$ and therefore $dist(p_v, q_v) \leq \rho$. A point $\tilde{p} \in \mathrm{Im}(\gamma_3)$ is either closer to p_v or q_v. Assuming, w.l.o.g, that it is closer to p_v we conclude that $dist(\tilde{p}, p_v) \leq \rho/2$. m_v is ρ-intersecting, thus $dist(p, p_v) \leq \sqrt{3}\rho/2$. As the segments (p, p_v) and $(p_v, \tilde{p})$ are perpendicular we apply the Pythagorean Theorem:

$$dist(p, p_0) = \sqrt{dist(p, p_v)^2 + dist(p_v, \tilde{p})^2} \leq \left(\frac{\sqrt{3}}{2}\rho\right)^2 + \left(\frac{\rho}{2}\right)^2 = \rho. \qquad \square$$

Theorem 1 *Let $a, b \in \mathcal{C}_{\mathrm{free}}$ such that there exists a collision-free path $\gamma_{a,b} \in \Gamma$ of length L and clearance ρ between a and b. Then the probability of the MMS algorithm to return a path between a and b after generating n_θ layers and n_p vertical lines is*:

$$\begin{aligned}\Pr\big[(a,b)\mathrm{SUCCESS}\big] &= 1 - \Pr\big[(a,b)\mathrm{FAILURE}\big] \\ &\geq 1 - \left\lceil \frac{L}{\rho} \right\rceil \Big[\big(1-\pi r^2\big)^{n_p} + (1-\rho)^{n_\theta}\Big],\end{aligned}$$

where $r = \frac{\sqrt{3}-1}{2}\rho$.

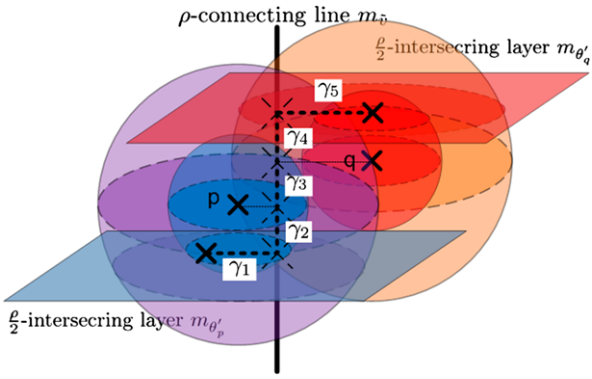


Fig. 6 Illustration of path induced by $\frac{\rho}{2}$-intersecting layers of p and q and a ρ-connecting line of p, q

Proof Let $l = \lceil \frac{L}{\rho} \rceil$, there exists a sequence $[p_0 \dots p_l]$ such that $p_i \in \mathrm{Im}(\gamma_{a,b})$, $p_0 = a$, $p_l = b$, $B_\rho(p_i) \in \mathcal{C}_{\text{free}}$ and $dist(p_i, p_{i+1}) \leq \rho$. *MMS* adds the manifolds m_{θ_a} and m_{θ_b} to the connectivity graph. m_{θ_a} has an FSC intersecting $B_{\frac{\rho}{2}}(a)$ and m_{θ_b} has an FSC intersecting $B_{\frac{\rho}{2}}(b)$.

Let $M'_P \subseteq M_P$ be the set of n_p lines sampled at random by the MMS algorithm and $M'_\Theta \subseteq M_\Theta$ be the set of n_θ layers sampled at random by the MMS algorithm. If there is a subset of the sampled lines $\{m_{\tilde{p}_1} \dots m_{\tilde{p}_{l-1}}\} \subseteq M'_P$ and a subset of the sampled layers $\{m_{\theta'_1} \dots m_{\theta'_{l-1}}\} \subseteq M'_\Theta$ such that $(m_{\theta'_i}, m_{\tilde{p}_i}, m_{\theta'_{i+1}})$ follow the conditions of Lemma 3 for $i \in \{0 \dots l-1\}$, then a path from $B_{\frac{\rho}{2}}(p_0)$ to $B_{\frac{\rho}{2}}(p_l)$ may be constructed. This is done by concatenating adjacent paths. Moreover, as $m_{\theta'_0} = m_{\theta_a}$ and $m_{\theta'_l} = m_{\theta_b}$, the path can start at a and terminate at b such that its distance from $\gamma_{a,b}$ is at most ρ hence it is in $\mathcal{C}_{\text{free}}$.

Let $I_0 \dots I_{l-1}$ be a set of indicator variables such that each I_i witnesses the event that there is a ρ-connecting vertical line $m_{\tilde{p}_i}$ of p_i and p_{i+1}. Let $J_1 \dots J_{l-1}$ be a set of indicator variables such that each J_i witnesses the event that there is a $\frac{\rho}{2}$-intersecting layer of p_i. It follows that MMS succeeds in answering the query (a, b) if $I_i = 1$ for all $0 \leq i \leq l-1$ and $J_j = 1$ for all $1 \leq j \leq l-1$. Therefore,

$$\begin{aligned}\Pr\big[(a,b)\text{FAILURE}\big] &\leq \Pr\big(\vee_{i=0}^{l-1}(I_i = 0) \vee_{i=1}^{l-1}(J_j = 0)\big) \\ &\leq \sum_{i=0}^{l-1} \Pr[I_i = 0] + \sum_{j=1}^{l-1} \Pr[J_j = 0].\end{aligned}$$

The events $I_i = 0$ are independent since the vertical samples are independent. The events $J_j = 0$ are independent since the layer samples are independent. $\Pr[I_i = 1] = \pi r^2$ for $r = \frac{\sqrt{3}-1}{2}\rho$. Thus the probability that none of the n_p uniform independent point samples falls in $D_r(\frac{\tilde{p}_{i+1}+\tilde{p}_i}{2})$ is $\Pr[I_i = 0] = (1 - \pi r^2)^{n_p}$ for $r = \frac{\sqrt{3}-1}{2}\rho$. $\Pr[J_j = 1] = \rho$. Thus the probability that none of the n_θ uniform independent angle samples falls in $[p_j - \frac{\rho}{2}, p_j + \frac{\rho}{2}]$ is $\Pr[J_j = 0] = (1 - \rho)^{n_\theta}$. Since the sampling is uniform and independent:

$$\Pr\big[(a,b)\text{FAILURE}\big] \le \left\lceil \frac{L}{\rho} \right\rceil \big(1-\pi r^2\big)^{n_p} + \left\lceil \frac{L}{\rho} - 1 \right\rceil (1-\rho)^{n_\theta}$$
$$\le \left\lceil \frac{L}{\rho} \right\rceil \big[\big(1-\pi r^2\big)^{n_p} + (1-\rho)^{n_\theta}\big]. \qquad \square$$

It follows that as n_p and n_θ tend to ∞, the probability of failing to find a path under the conditions stated in Theorem 1 tends to zero.

5 Implementation and Performance

This section gives further background and details of our implementation and compares this implementation with two versions of the PRM algorithm, which are implemented in OMPL [12]: the standard version using uniform sampling [22]; and another one using obstacle based sampling [1].

5.1 Implementation Details

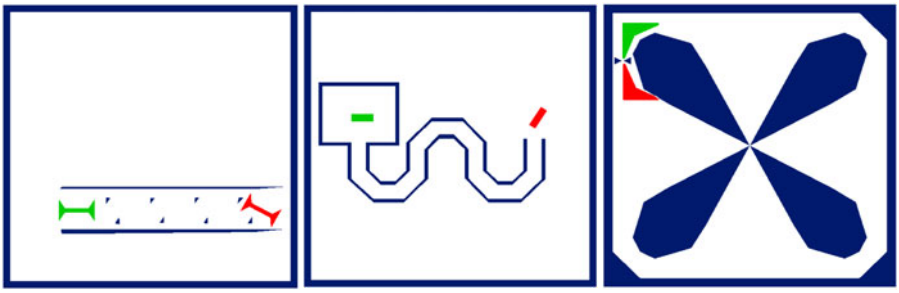
The algorithm discussed in Sect. 3 is implemented in C++. It is based on CGAL's arrangement package, which is used for the geometric primitives, and the BOOST graph library [39], which is used to represent the connectivity graph $\mathcal{G}$. We next discuss the manifold decomposition methods in more detail.

Angle-Primitive The Angle-Primitive for a constraining angle θ (denoted $P_\Theta(\theta)$) is constructed by computing the Minkowski sum of $-R_\theta$ with the obstacles, where $-R_\theta$ denotes R rotated by θ and reflected about the origin. The implementation is an application of CGAL's Minkowski sums package [40, C.24]. We remark that we ensure (using the method of Canny et al. [8]) that the angle θ is chosen such that $\sin\theta$ and $\cos\theta$ are rational. This allows for an exact rotation of the robot and an exact computation of the Minkowski Sum.

Segment-Primitive Limiting the possible positions of the robot's reference point r to a given segment s, results in a two-dimensional configuration space. Each vertex (or edge) of the robot in combination with each edge (or vertex) of an obstacle gives rise to a critical curve in this configuration space. Namely the critical curves are the set of all configurations that put the two features into contact, and thus mark a potential transition between $\mathcal{C}_{\text{forb}}$ and $\mathcal{C}_{\text{free}}$. In this particular setting it turns out that all critical curves can be expressed as rational functions only. We therefore added exact support for rational functions to CGAL, which is available in CGALrelease 3.9 and higher. Compared to the currently fastest alternative, that of general algebraic curves [6], we gained a speed up factor of about 4. For a full discussion of the occurring critical curves as well as further implementation details we refer the reader to [35].

5.2 Experimental Results

We demonstrate the performance of our planner using three different scenarios. All scenarios consist of a workspace, a robot with obstacles and one query (source and



(a) Tunnel scenario (b) Snake scenario (c) Flower scenario

Fig. 7 Experimental scenarios

Table 1 Parameter sensitivity for Flower Scenario

n_s	n_θ			
	20	40	80	160
64	(1.3,×)	(4.1,✓)	(12.2,✓)	(60,✓)
128	(1.3,×)	(4.1,✓)	(13.2,✓)	(63,✓)
256	(1.4,×)	(4,✓)	(13.5,✓)	(65,✓)
512	(1.3,×)	(4,✓)	(14.8,✓)	(79,✓)

target configurations). Figure 7 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 7 runs excluding the minimal and maximal values.

5.2.1 Algorithm Properties

Our planner has two parameters: the number n_θ of layers to be generated and the number n_s of segment constraints to be generated. We chose the following values for these parameters: $n_\theta \in \{10, 20, 40, 80, 160, 320\}$ and $n_s \in \{2^i \mid i \in \mathbb{N}, i \leq 14\}$. For a set of parameters (n_θ, n_s) we report the preprocessing time t (in seconds) and whether a path was found (marked ✓) or not found (marked ×) once the query was issued. The results for the flower scenario are reported in Table 1. We show that a considerable increase in parameters has only a limited effect on the preprocessing time.

In order to test the effectiveness of our optimizations, we ran the planner with and without any heuristic for choosing segments and with and without segment filtering. The results for the flower scenario can be viewed in Table 2. We remark that the engineering work invested in optimizing MMS yielded an algorithm comparable and even surpassing a motion planner that is in prevalent use as shown next.

Table 2 Optimization results for Flower Scenario. Two optimizations were considered: The first, is the method to generate segments for the segment primitive, either by random segments or using the heuristic of attempting to connect overlapping FSC's of adjacent layers. The second optimization is if to apply constraint filtering, i.e, to discard constraints that will intersect FSC's belonging to the same connected component of the connectivity graph

Segment generation	Constrained filtering	n_θ	n_s	t [s]
random	not used	40	1024	255
	used	40	1024	25
heuristic	not used	40	64	22
	used	40	256	4

Table 3 Comparison with PRM and OBPRM

Scenario	MMS			PRM			OBPRM		
	n_θ	n_s	t [s]	k	*res*	t [s]	k	*res*	t [s]
Tunnel	20	128	21	10	0.005	114	10	0.01	134
Snake	40	256	35	10	0.02	264	10	0.01	247
Flower	40	256	4	20	0.02	20	14	0.01	20

5.2.2 Comparison with PRM

We used the open motion planning library (OMPL) [12] for the implementations of the PRM algorithm. We chose to compare our planner to two variants of the PRM: The first is the standard implementation where new samples are sampled uniformly from the configuration space (we denote this variant simply as PRM). The second, obstacle-based PRM or OBPRM [1], biases samples near the boundary of obstacles. OBPRM was 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. As with previous tests, the parameters for MMS are n_θ and n_s. The parameters used for the PRM are the number of neighbors (denoted k) to which each milestone should be connected and collision checking resolution (denoted *res* in Table 3).

Furthermore, we ran the flower scenario several times, progressively increasing the robot size. This caused a "tightening" of the passages containing the desired path. Figure 8 demonstrates the preprocessing time as a function of the tightness of the problem for all planners. A tightness of zero denotes the base scenario (Fig. 7c) while a tightness of one denotes the tightest problem solved.

The results show a speedup for all scenarios when compared to the PRM implementations. Moreover, our algorithm has little sensitivity to the tightness of the problem as opposed to the PRM implementations. In the tightest experiment solved by the both PRM variants, MMS runs more than 58 times faster. We ran the experiment on a tighter scenario where both the PRM variants terminated due to insufficient memory, our algorithm's running time changed little for this scenario.

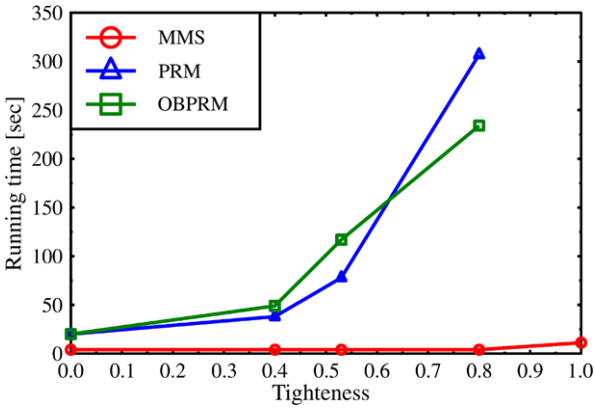


Fig. 8 Tightness results for the flower scenario

6 Further Directions

Our primary goal is to use the MMS framework to solve progressively more complicated motion-planning problems. As suggested earlier, we see the framework as a platform for convenient transfer of strong geometric primitives into motion planning algorithms. For example, among the recently developed tools are efficient and exact solutions for computing the Minkowski sums of polytopes in $\mathbb{R}^3$ [15, 17, 42], for exact update of the sum when the polytopes rotate [31], and for the exact construction of the configuration space of a rotating polyhedron [14]. These could be combined into an MMS for planning full rigid motion of a polytope among polytopes, which extrapolating from the current experiments could outperform more simplistic solutions in existence. We also wish to extend the scheme to the case of single query planning. We hope that these future efforts will help to establish the advantages of the framework over existing methods.

As part of our ongoing work we were able to report on the following results in [34]: (i) 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 PRM, with over 20-fold speedup in a coordination-tight setting. (ii) A novel characterization of narrow passages that attempts to capture their dimensionality. (iii) Insights about the advantageous behavior of our approach in the presence of *high-dimensional narrow passages*. (iv) A probabilistic completeness proof for the most prevalent case, namely MMS with samples that are affine subspaces. However, it would be interesting to formalize the characteristics described in Sect. 2.1 in a way that could lead to a more general proof of probabilistic completeness.

Looking at more intricate problems, we anticipate some difficulty in turning constraints into manifolds that can be exactly and efficiently decomposed. We propose to have manifolds where the decomposition yields some *approximation* of the FSCs, using recent advanced meshing tools for example. Additionally, one may use approximate constructions of the Minkowski sum of two polyhedral models as presented by Lien [29] and Varadhan and Manocha [41]. We can endow the connectivity-graph

nodes with an attribute describing their approximation quality. One can then decide to only look for paths all whose nodes are above a certain approximation quality. Alternatively, one can extract any solution path and then refine only those portions of the path that are below a certain quality.

Beyond Motion Planning We foresee an extension of the framework to other problems that involve high-dimensional arrangements of critical hypersurfaces. It is difficult to describe the entire arrangement analytically, but there are often situations where constraint manifolds could be computed analytically. Hence, it is possible to shed light on problems such as loop closure and assembly planning where we can use manifold samples to analytically capture pertinent information of high-dimensional arrangements of hypersurfaces. Notice that although in Sect. 3 we used only planar manifolds, there are recently developed tools to construct two-dimensional arrangement of curves on curved surfaces [5], which gives further flexibility in choosing the manifold families.

For supplementary material the reader is referred to our webpage http://acg.cs.tau.ac.il/projects/mms.

References

1. Amato, N.M., Wu, Y.: A randomized roadmap method for path and manipulation planning. In: IEEE International Conference on Robotics and Automation, pp. 113–120 (1996)
2. Aronov, B., Sharir, M.: On translational motion planning of a convex polyhedron in 3-space. SIAM J. Comput. **26**(6), 1785–1803 (1997)
3. Avnaim, F., Boissonnat, J., Faverjon, B.: A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles. In: Geometry and Robotics. Lecture Notes in Computer Science, vol. 391, pp. 67–86. Springer, Berlin (1989)
4. Basu, S., Pollack, R., Roy, M.F.: Algorithms in Real Algebraic Geometry, 2nd edn. Springer, Secaucus (2006)
5. Berberich, E., Fogel, E., Halperin, D., Mehlhorn, K., Wein, R.: Arrangements on parametric surfaces I: General framework and infrastructure. Math. Comput. Sci. **4**(1), 45–66 (2010)
6. Berberich, E., Hemmer, M., Kerber, M.: A generic algebraic kernel for non-linear geometric applications. In: Symposium on Computational Geometry, pp. 179–186 (2011)
7. Berenson, D., Srinivasa, S.S., Ferguson, D., Kuffner, J.J.: Manipulation planning on constraint manifolds. In: International Conference on Robotics and Automation, pp. 625–632 (2009)
8. Canny, J., Donald, B., Ressler, E.K.: A rational rotation method for robust geometric algorithms. In: Symposium on Computational Geometry, pp. 251–260 (1992)
9. Canny, J.F.: Complexity of Robot Motion Planning (ACM Doctoral Dissertation Award). MIT Press, Cambridge (1988)
10. Chazelle, B., Edelsbrunner, H., Guibas, L.J., Sharir, M.: A singly exponential stratification scheme for real semi-algebraic varieties and its applications. Theor. Comput. Sci. **84**(1), 77–105 (1991)
11. Choset, H., Burgard, W., Hutchinson, S., Kantor, G., Kavraki, L.E., Lynch, K., Thrun, S.: Principles of Robot Motion: Theory, Algorithms, and Implementation. MIT Press, Cambridge (2005)
12. Şucan, I.A., Moll, M., Kavraki, L.E.: The open motion planning library. IEEE Robotics & Automation Magazine (2012). http://ompl.kavrakilab.org
13. De Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Berlin (2008)
14. Dobrowolski, P.: An algorithm for computing the exact configuration space of a rotating object in 3-space. Int. J. Comput. Sci. **39**(4), 363–376 (2012)
15. Fogel, E., Halperin, D.: Exact and efficient construction of Minkowski sums of convex polyhedra with applications. Comput. Aided Des. **39**(11), 929–940 (2007)

16. Fogel, E., Halperin, D., Wein, R.: CGAL Arrangements and Their Applications—A Step-by-Step Guide, Geometry and Computing vol. 7. Springer, Berlin (2012)
17. Hachenberger, P.: Exact Minkowski sums of polyhedra and exact and efficient decomposition of polyhedra into convex pieces. Algorithmica **55**(2), 329–345 (2009)
18. Halperin, D., Sharir, M.: A near-quadratic algorithm for planning the motion of a polygon in a polygonal environment. Discrete Comput. Geom. **16**(2), 121–134 (1996)
19. Hirsch, S., Halperin, D.: Hybrid motion planning: Coordinating two discs moving among polygonal obstacles in the plane. In: Workshop on the Algorithmic Foundations of Robotics, pp. 225–241 (2002)
20. Hsu, D., Latombe, J.C., Motwani, R.: Path planning in expansive configuration spaces. Int. J. Comput. Geom. Appl. **9**(4/5), 495–512 (1999)
21. Kavraki, L.E., Kolountzakis, M.N., Latombe, J.C.: Analysis of probabilistic roadmaps for path planning. IEEE Trans. Robot. Autom. **14**(1), 166–171 (1998)
22. Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high dimensional configuration spaces. IEEE Trans. Robot. Autom. **12**(4), 566–580 (1996)
23. Kuffner, J.J., Lavalle, S.M.: RRT-Connect: An efficient approach to single-query path planning. In: IEEE International Conference on Robotics and Automation, pp. 995–1001 (2000)
24. Ladd, A.M., Kavraki, L.E.: Generalizing the analysis of PRM. In: IEEE International Conference on Robotics and Automation, pp. 2120–2125 (2002)
25. Latombe, J.C.: Robot Motion Planning. Kluwer Academic, Norwell (1991)
26. Lavalle, S.M.: Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Computer Science Department, Iowa State University (1998)
27. LaValle, S.M.: Planning Algorithms. Cambridge University Press, Cambridge (2006)
28. Lien, J.M.: Hybrid motion planning using Minkowski sums. In: Robotics: Science and Systems (2008)
29. Lien, J.M.: A simple method for computing Minkowski sum boundary in 3D using collision detection. In: Workshop on the Algorithmic Foundations of Robotics, pp. 401–415 (2008)
30. Lozano-Perez, T.: Spatial Planning: A Configuration Space Approach. MIT Press, Cambridge (1980). AI Memo 605
31. Mayer, N., Fogel, E., Halperin, D.: Fast and robust retrieval of Minkowski sums of rotating convex polyhedra in 3-space. In: SPM, pp. 1–10 (2010)
32. Porta, J.M., Jaillet, L., Bohigas, O.: Randomized path planning on manifolds based on higher-dimensional continuation. Int. J. Robot. Res. **31**(2), 201–215 (2012)
33. Reif, J.H.: Complexity of the mover's problem and generalizations. In: Symposium on Foundations of Computer Science, pp. 421–427. IEEE Comput. Soc., Washington (1979)
34. Salzman, O., Hemmer, M., Halperin, D.: On the power of manifold samples in exploring configuration spaces and the dimensionality of narrow passages. In: Workshop on the Algorithmic Foundations of Robotics (2012, to appear). arXiv:1202.5249
35. Salzman, O., Hemmer, M., Raveh, B., Halperin, D.: Motion planning via manifold samples. In: European Symposium on Algorithms, pp. 493–505 (2011)
36. Schwartz, J.T., Sharir, M.: On the "piano movers" problem: I. The case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. Commun. Pure Appl. Math. **35**, 345–398 (1983)
37. Schwartz, J.T., Sharir, M.: On the "piano movers" problem: II. General techniques for computing topological properties of real algebraic manifolds. Adv. Appl. Math. **4**(3), 298–351 (1983)
38. Sharir, M.: Algorithmic Motion Planning, Handbook of Discrete and Computational Geometry, 2nd edn. CRC Press, Boca Raton (2004)
39. Siek, J.G., Lee, L.Q., Lumsdaine, A.: The Boost Graph Library: User Guide and Reference Manual. Addison-Wesley, Reading (2001)
40. The CGAL Project: CGAL User and Reference Manual, 3.7 edn. CGAL Editorial Board (2010). http://www.cgal.org/
41. Varadhan, G., Manocha, D.: Accurate Minkowski sum approximation of polyhedral models. Graph. Models **68**(4), 343–355 (2006)
42. Wein, R.: Exact and efficient construction of planar Minkowski sums using the convolution method. In: European Symposium on Algorithms, pp. 829–840 (2006)
43. Yang, J., Sacks, E.: RRT path planner with 3 DOF local planner. In: IEEE International Conference on Robotics and Automation, pp. 145–149 (2006)
44. Zhang, L., Kim, Y.J., Manocha, D.: A hybrid approach for complete motion planning. In: International Conference on Intelligent Robots and Systems, pp. 7–14 (2007)