Narrow Passage Manipulation Planning using Homotopy Optimization Method

Minji Lee[1], Jeongmin Lee[2] and Dongjun Lee[1,†]
[1] Seoul National University, [2] Holliday Robotics

***Abstract*— Manipulation in a narrow passage is a common challenge in both industrial and household environments, often resulting in infeasible solutions or high computational cost. To address the risk of optimization failure caused by narrow passage, we propose a homotopy optimization framework that maintains feasibility throughout the process via a sequence of easier subproblems specifically designed for narrow passage scenarios. The approach begins by decomposing the environment into convex objects and initializing collision constraints using only a subset of these objects. The remaining obstacles are then introduced progressively through an interpolation of signed distance–based collision constraints. This process yields a series of intermediate subproblems that gradually guide the solution toward the final, fully constrained problem, while avoiding infeasible states. We demonstrate the effectiveness of the framework through a set of examples, showing its ability to reliably solve manipulation planning problems in environments with narrow passages.**

I. Introduction

Manipulation in constrained or cluttered environments frequently demands solving difficult path planning problems, especially through narrow passages [1], [2], [3], [4]. These challenges arise in a wide range of tasks, from structured household manipulation to operations in industrial workspaces. Sampling-based planners such as PRM [5] and RRT [6] are easy to formulate and offer probabilistic completeness [7], but their sampling efficiency degrades significantly in narrow environments, making it difficult to find feasible paths within a reasonable time. Optimization-based planners such as CHOMP [8] and TrajOpt [9], support highly flexible problem formulations that can incorporate a wide range of costs and constraints, and usually converges rapidly. However, if the waypoints penetrate obstacles deeply, the optimization turns ill-posed, and the solver can stall in an infeasibility [10].

To overcome both limitations we adopt a homotopy–based strategy. Homotopy optimization method [11] starts from a simplified problem that is easy to solve and then incrementally restores the original complexity while continuously refining the solution. In our setting the simplification is achieved by relaxing collision constraints so that the initial feasible space is wide; as the algorithm proceeds we gradually tighten these constraints, steering the path toward a valid solution without ever allowing deep penetration to corrupt the optimization process.

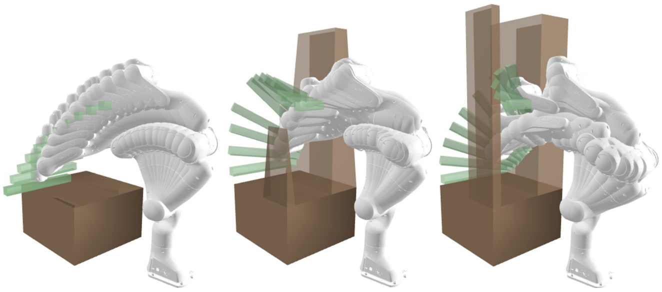

Fig. 1: Optimization results for manipulator path planning during tool extraction from a narrow gap, utilizing the proposed collision constraint interpolation framework.

II. Collision-Constraint Interpolation

Our goal is to keep the path feasible throughout homotopy optimization process, thereby sidestepping the deep penetration problem. Let the environment be decomposed into a set of convex objects $\mathcal{V}_{tot} = \{v_1, \cdots, v_{n_{tot}}\}$. Beginning with an environment initialized with a subset of the convex objects $\mathcal{V}_{init} \subseteq \mathcal{V}_{tot}$, we introduce the others in batches that satisfy:

Condition 1 (Leaf set): A set $\mathcal{V}^\ell$ with respect to $\mathcal{V}$ is a *leaf set* if

1) every $v^\ell \in \mathcal{V}^\ell$ intersects *exactly one* object in $\mathcal{V}$, and
2) no two elements of $\mathcal{V}^\ell$ intersect each other.

Note that attaching a leaf set $\mathcal{V}^\ell$ to $\mathcal{V}$ cannot create or destroy cycles, thereby the homotopy equivalence is preserved.

A. SDF Interpolation between Convex Objects

Let v_1 and v_2 be intersecting convex objects. We interpolate their signed distance functions (SDFs) by

$$\mathrm{sd}^{\alpha}_{v_1\to v_2}(x) = (1-\alpha)\, f\big(\mathrm{sd}_{v_1}(x)\big) + \alpha\, f\big(\mathrm{sd}_{v_2}(x)\big) \qquad (1)$$

where $\mathrm{sd}_v(\cdot) : \mathbb{R}^3 \to \mathbb{R}$ is a SDF of an object v, $\alpha \in [0,1]$ is an interpolation variable, and $f(\cdot) : \mathbb{R} \to \mathbb{R}$ is an *exponential shaping function*, defined as:

$$f(x) = \frac{e^{\eta x} - 1}{\eta}, \qquad \eta > 0, \qquad (2)$$

Because f is convex and non-decreasing, and each sd_{v_i} is convex [12], the composite (1) is convex for every α. Consequently its sub-zero-level set

$$v^{\alpha}_{v_1\to v_2} := \big\{x \mid \mathrm{sd}^{\alpha}_{v_1\to v_2}(x) \le 0\big\}$$

This research was supported by Samsung Research, the National Research Foundation (NRF) funded by the Ministry of Science and ICT (MSIT) of Korea (RS-2022-00144468), and the Ministry of Trade, Industry & Energy (MOTIE) of Korea (RS-2024-00419641).

Minji Lee and Dongjun Lee are with the Department of Mechanical Engineering, IAMD and IOER, Seoul National University, Seoul, Republic of Korea, 08826. {mingg8,ljmlgh,djlee}@snu.ac.kr. Corresponding author: Dongjun Lee.

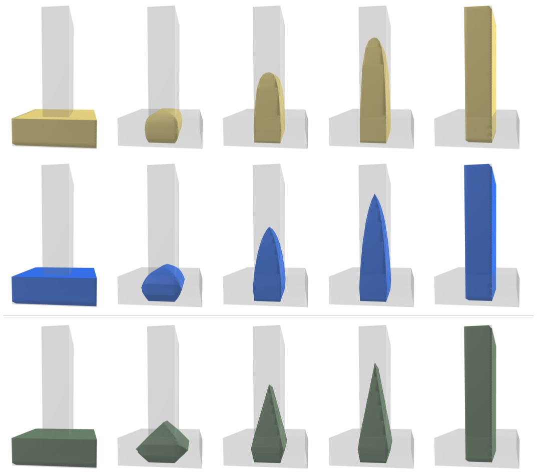

Fig. 2: Object interpolation process using proposed shaping function (2) with $\eta = 40$ (top row), $\eta = 15$ (middle row) and linear interpolation $\eta \to 0$ (bottom row).

is a convex object satisfying $v_1 \cap v_2 \subseteq v^{\alpha}_{v_1 \to v_2} \subseteq v_1 \cup v_2$. The exponential shaping suppresses the sharp ridges that a linear blend ($\eta = 0$) would generate along the medial axis, thereby preventing abrupt flips of contact normals that hamper gradient-based optimization, as shown in Fig. 2

B. Homotopy-Preserving Addition of Leaf Set

For a k-th batch, we consider a set $\mathcal{V}^{\ell}_k = \{v^{\ell}_j\} \subseteq \mathcal{V}_{tot}$ which is a leaf set with respect to a current set $\mathcal{V}_k$. We then interpolate their SDFs as:

$$\mathrm{sd}^{\alpha}_{\mathcal{V}_k + \mathcal{V}^{\ell}_k} = \min\bigl(\mathrm{sd}_{\mathcal{V}_k}, \mathrm{sd}^{\alpha}_1(x), \cdots, \mathrm{sd}^{\alpha}_{n_\ell}(x)\bigr),$$

where $\mathrm{sd}^{\alpha}_j(x) := \mathrm{sd}^{\alpha}_{v_{\sigma(j)} \to v^{\ell}_j}(x)$, and $v_{\sigma(j)} \in \mathcal{V}_k$ is the unique convex object that intersects with v^{ℓ}_j (Condition 1-1). The resulting occupied space can be written as:

$$\mathcal{O}\bigl(\mathrm{sd}^{\alpha}_{\mathcal{V}_k + \mathcal{V}^{\ell}_k}\bigr) = \mathcal{O}_{\mathcal{V}_k} \cup \bigcup_{j=1}^{n_\ell} \underbrace{\Bigl\{ v^{\alpha}_{v_{\sigma(j)} \to v^{\ell}_j} \cap v^{\ell}_j \Bigr\}}_{=: v^{\alpha}_j}$$

where $\mathcal{O}(\cdot)$ denotes the zero–sub-level set of a function. This expression corresponds to gluing a new leaf set consists of $v^{\alpha}_1, \cdots, v^{\alpha}_{n_\ell}$ to the existing environment. Since such an operation preserves topological features, the interpolation maintains the homotopy type of $\mathcal{O}_{\mathcal{V}_k}$ for all $\alpha \in [0, 1]$.

III. Manipulation Path Planning using Collision Constraint Interpolation

A typical path planning optimization can be formulated as:

$$\begin{aligned} \underset{X_{1:T}}{\text{minimize}} \quad & \sum_{t=1}^{T-1} \bigl\| X_{t+1} - X_t \bigr\|^2 + f_i(X_1) + f_g(X_T) \\ \text{subject to} \quad & \min_{x \in W(X_t)} \mathrm{sd}_{\mathcal{V}_{tot}}(x) \geq \hat{d}, \; t = 1, \ldots, T, \end{aligned} \tag{3}$$

where $X_{1:T}$ is the optimization variable representing configuration of waypoints, T is the total number of waypoints, f_i and f_g is a cost function for initial and goal waypoints, $W(X_t)$ is the surface of the robot at each waypoint X_t, and $\hat{d} \in \mathbb{R}^+$ is the predefined safe distance.

Algorithm 1 Narrow passage path planning

```
**Input**: Total set of nodes $\mathcal{V}_{tot}$
Determine $\mathcal{V}^{\ell}, \mathcal{V}_{init}$ by identifying leaf sets
Initialize environment $\mathcal{V}_1 = \mathcal{V}_{init}$ and path $X_{1:T}$
for $k = 1, \cdots, |\mathcal{V}^{l}|$ do
    $\alpha = 0$
    while $\alpha < 1$ do
        Refine the path $X_{1:T}$ by optimization (4)
        $\alpha \leftarrow \min(\alpha + \Delta\alpha, 1)$
    end while
    $\mathcal{V}_{k+1} = \mathcal{V}_k \cup \mathcal{V}^{\ell}_k$
end for
return $\mathcal{V}_{seq}$
```

To cope with challenges of narrow passages, we replace the collision avoidance constraint in (3) by a sequence of progressively harder sub-problems obtained through the collision constraint interpolation of Sec. II. Each subproblem keeps the same objective function but enforces relaxed constraints:

$$\min_{x \in W(X_t)} \mathrm{sd}^{\alpha}_{\mathcal{V}_k + \mathcal{V}^{\ell}_k}(x) \geq \hat{d} \tag{4}$$

with the interpolation variable α rising from 0 (wide free space) to 1 (exact geometry) while leaf sets $\mathcal{V}^{\ell}_k$ are successively attached to the current environment $\mathcal{V}_k$. Solving these subproblems in order yields a path that remains feasible throughout the homotopy and ends in a collision-free trajectory for the original, unrelaxed problem. The overall path planning algorithm is explained in Algorithm 1.

To solve the subproblem (4), we employ Sequential Quadratic Programming (SQP). The resulting Quadratic Programming (QP) at each iteration is solved using SubADMM [13], which is particularly adept at stably and efficiently solving conflicting constraints common in narrow passage path planning.

IV. Evaluations

We test our framework in two different manipulation problems, 1) placing dishes on a drying rack, 2) extracting a box from a narrow gap. The scenarios are compared against the results obtained using baseline planners from TrajOpt [9], OMPL [14] and CHOMP [8]. The timeout for the OMPL was set to 30 seconds and 50 seconds for each respective scenario.

A. Placing Dishes on the Rack

Inserting a dish into a narrow gap of a drying rack is challenging for a manipulator [15]. The thinness of the rack makes deep penetration more likely, resulting in being stuck in infeasibility. For the same reason, even achieving the feasible goal position is challenging for this scenario.

We define the objective function as the distance to an approximate reference pose located at the center of the rack,

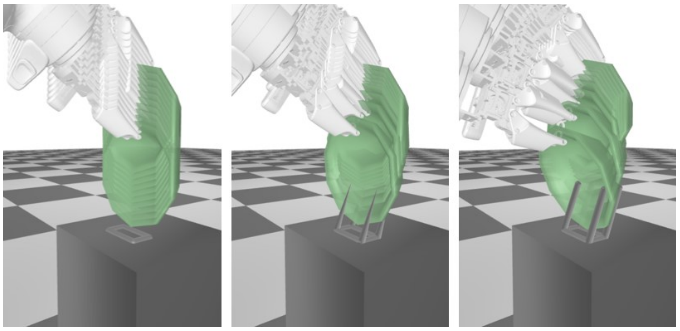

Fig. 3: Results of a successful path planning using the proposed framework for the dish insertion task.

facing horizontal direction. Additionally, the Cartesian path length objective function and a hard constraint on the initial joint position are incorporated. By employing our proposed initialization scheme and refinement process, feasible final position of the plate placed on the drying rack, along with a feasible path could be achieved.

Table I compares the result of the optimization with and without collision constraint interpolation. Success time shows the mean and standard deviation of the elapsed time of success cases, while total time shows both success and failure cases. The tests are conducted using combinations of three different shapes of dishes and 20 different racks. The results indicate that our method outperforms the one without collision constraint interpolation in both success rate and computation time.

Method	**Success**	**Total time (s)**	**Success time (s)**
Proposed	58/60	3.56 ± 0.69	3.95 ± 0.68
Without interpolation	23/60	7.06 ± 0.33	7.42 ± 0.27

TABLE I: Ablation study of dish placing with and without collision constraint interpolation.

Method	**Success**	**Total time (s)**	**Success time (s)**
Proposed	20/20	3.28 ± 0.61	3.28 ± 0.61
RRTConnect	9/20	20.08 ± 11.18	10.45 ± 10.32
BiTRRT	10/20	22.21 ± 9.91	16.24 ± 11.10
TRRT	2/20	26.37 ± 9.02	1.36 ± 0.00
BiEST	4/20	23.42 ± 9.91	16.24 ± 11.10
BMFT	2/20	21.34 ± 7.86	10.85 ± 8.33
PRMstar	0/20	31.67 ± 0.90	-
LazyPRM	0/20	30.03 ± 0.00	-
KPIECE	1/20	22.20 ± 9.09	8.48 ± 0
BKPIECE	4/20	21.34 ± 7.86	10.85 ± 8.33
TrajOpt	0/20	4.76 ± 1.27	-
CHOMP	2/20	22.74 ± 2.58	15.28 ± 0.03

TABLE II: Comparison with other methods for the task of placing dish, showing the planning time and success rate, with the goal position provided by our framework.

Conventional planning methods typically require a predefined goal pose. To compare our proposed framework with existing methods, we used the feasible final pose obtained by our framework as the goal position for the other methods. We tested twelve different shapes of dish racks, with the results presented in Table II. As also shown in [2], conventional sampling-based methods—except for BiTRRT [16]—faced significant challenges in solving the narrow passage problem. Despite making the problem easier for the baselines by providing the goal pose, our method still outperformed all others in both success rate and computation time. Note that in the case of TrajOpt, continuous collision detection is performed by assuming a convex hull between waypoints. However, this results in an overly conservative over-approximation for the non-convex geometry of the dish, leading to failure in successful execution.

Method	**Success**	**Total time (s)**	**Success time (s)**
Proposed	30/30	6.52 ± 0.69	6.52 ± 0.69
RRTConnect	1/30	48.83 ± 3.54	32.91 ± 0.00
BiTRRT	19/30	30.33 ± 14.23	23.15 ± 12.80
TRRT	0/30	50.03 ± 0.00	-
BiEST	0/30	50.07 ± 0.16	-
BMFT	1/30	49.51 ± 4.98	32.41 ± 0.00
PRMstar	0/30	50.28 ± 0.41	-
LazyPRM	0/30	50.04 ± 0.01	-
KPIECE	0/30	50.04 ± 0.015	-
BKPIECE	0/30	50.07 ± 0.14	-
TrajOpt	17/30	3.41 ± 1.09	2.71 ± 0.93
CHOMP	0/30	32.66 ± 11.20	-

TABLE III: Comparison with other methods for the task of extracting tool from a narrow gap, showing the planning time and success rate.

B. *Extraction of Tool from a Narrow Gap*

Taking tool out through a narrow gap is a challenge task for a manipulator, especially when the size of the tool is large, or the obstacles are thin. Our objective is to optimize the manipulator path, starting from a pose that grasping the object, and extracting it out through a narrow gap. Fig 1 shows the result of the optimization.

We introduced slight randomness into the environment configuration to create 30 environments. Each planner was tested under these conditions. As shown in Table III, CHOMP and sampling methods had low success rates and longer planning times compared to our approach. While TrajOpt achieved higher success rates and shorter planning times than the sampling methods, its success rates were still lower than our framework, which successfully planned in all configurations.

V. Conclusions

The proposed framework leverages a homotopy optimization strategy to effectively handle narrow passage manipulation planning, but several limitations remain. First, new obstacles can be introduced only when they form a leaf set; environments whose topology violates this condition fall outside the method's scope. Second, the need for a full convex decomposition adds preprocessing overhead, and the repeated update of the interpolation parameter α increases runtime in scenes that do not actually contain extreme bottlenecks. Third, like any optimization approach, the solver may converge to undesirable local minima. Future research will study alternative homotopy schedules, relax the leaf-set requirement to cover more intricate workspaces, and integrate global exploration strategies to mitigate local-minimum failure.

References

[1] S. Ruan, K. L. Poblete, H. Wu, Q. Ma, and G. S. Chirikjian. Efficient path planning in narrow passages for robots with ellipsoidal components. *IEEE Transactions on Robotics*, 39(1):110–127, 2022.

[2] S. Li and N. T. Dantam. Sample-driven connectivity learning for motion planning in narrow passages. In *IEEE International Conference on Robotics and Automation*, 2023.

[3] A. Orthey and M. Toussaint. Section patterns: Efficiently solving narrow passage problems in multilevel motion planning. *IEEE Transactions on Robotics*, 37(6):1891–1905, 2021.

[4] N. Hiraoka, H. Ishida, T. Hiraoka, K. Kojima, K. Okada, and M. Inaba. Sampling-based global path planning using convex polytope approximation for narrow collision-free space of humanoid. *International Journal of Humanoid Robotics*, page 2450005, 2024.

[5] L. E. Kavraki, P. Svestka, J-C Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.

[6] S. LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998.

[7] L. E. Kavraki, M. N. Kolountzakis, and J-C Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and automation*, 14(1):166–171, 1998.

[8] N. D. Ratliff, M. Zucker, J. A. Bagnell, and S. S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation*, pages 489–494, 2009.

[9] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel. Motion planning with sequential convex optimization and convex collision checking. *The International Journal of Robotics Research*, 33(9):1251–1270, 2014.

[10] K. Erleben. Methodology for assessing mesh-based contact point methods. *ACM Transactions on Graphics*, 37(3):1–30, 2018.

[11] D. M Dunlavy and D. P O'Leary. Homotopy optimization methods for global optimization. Technical report, Sandia National Laboratories (SNL), Albuquerque, NM, and Livermore, CA ..., 2005.

[12] S. Yan, X.-C. Tai, J. Liu, and H.-Y. Huang. Convexity shape prior for level set-based image segmentation method. *IEEE Transactions on Image Processing*, 29:7141–7152, 2020.

[13] J. Lee, M. Lee, and D. Lee. Modular and parallelizable multibody physics simulation via subsystem-based admm. In *IEEE International Conference on Robotics and Automation*, 2023.

[14] D. Coleman, I. Sucan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

[15] J. Lee, M. Lee, and D. Lee. Uncertain pose estimation during contact tasks using differentiable contact features. *Robotics: Science and Systems*, 2023.

[16] D. Devaurs, Thierry Siméon, and Juan Cortés. Enhancing the transition-based rrt to deal with complex cost spaces. In *IEEE international conference on robotics and automation*, pages 4120–4125, 2013.