Efficient $SE(3)$ Reachability Map Generation via Interplanar Integration of Intra-planar Convolutions

Yiheng Han[1], Jia Pan[2], Mengfei Xia[1], Long Zeng[3] and Yong-Jin Liu[1†]

***Abstract*— Convolution has been used for fast computation of reachability maps, but it has high computational costs when performing $SE(3)$ convolution operations for general joint arrangements in industrial robots and 3D workspace. Its application is also limited to planar robots, 2D workspace, or robots with special spatial arrangements for joints. In this paper, we find that the $SE(3)$ convolution can be decomposed into a set of $SE(2)$ convolutions, which significantly reduces the computational complexity when computing the reachability map of high-DOF robotic manipulators in the 3D workspace. We also leverage GPU parallel computing and Fast Fourier transform to further accelerate the computation procedure. We demonstrate the time efficiency and quality of our approach using a set of numerical experiments for constructing reachability maps and also present a multi-robot plant phenotyping system that uses the computed reachability map for efficient viewpoint selection and path planning.**

I. INTRODUCTION

The reachability map provides a structured description of a robot's ability to grasp and manipulate objects by taking into account the kinematic structure of a robotic arm [1]. The offline computed reachability map can help to save time for many online queries such as grasp selection, path planning [2], obstacle avoidance [3], [4], robot Cooperation and analysis [5], [6], and can also assist in determining the setup parameters such as the robot base position for optimizing the performance of predefined manipulation tasks [7], [8].

The reachability map is a 6-dimensional map indicating whether the robot's end-effector can reach a position (x, y, z) with an orientation of (α, β, γ). The map is generally implemented by dividing the 3D workspace into voxels, and each voxel is associated with a rotational grid that discretizes $SO(3)$. Each grid cell has an associated quality index counting the different joint configurations that allow the robot's end-effector to reach the pose $(x, y, z, \alpha, \beta, \gamma)$ [9]. A straightforward method for the reachability map generation uses a combination of forward kinematics (FK) and sampling in the joint configuration space. A brute-force uniform sampling has an exponential complexity of K^P, where K is the sampling resolution for each joint and P is the number of joints. To reduce the computational cost, random sampling can be used but it cannot guarantee that all the map cells can be covered in finite time. Inverse kinematics (IK) can also be used for the map generation and it guarantees a complete exploration of the 6-dimension cells. However, the IK solver takes longer time than the FK solver, and generally it only provides a binary indication about whether a cell is reachable or not, rather the set of all valid configurations satisfying the kinematic constraint. A visual demo is shown in Figure 1.

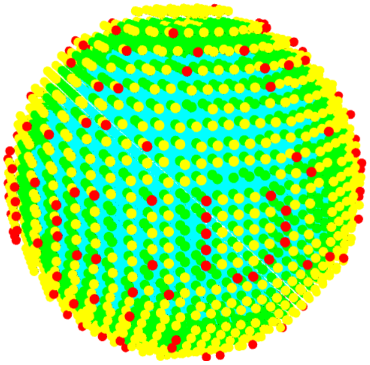


Fig. 1. A 3D visual demo to show the reachability map which is generated by UR manipulator. To facilitate visualization, reachability is added to 3-dimensional position. Different colors represent different reachability which is reduced in order of red, yellow, green and cyan.

To resolve the limitations of previous works, Chirikjian et al. [9], [10], [1], [11] proposed to formulate the reachability map computation as a series of convolutions on $SE(D)$ (where $D = 1$ or 2 is the dimension of the workspace), which can be solved by an elegant and efficient numerical algorithm. In this way, for a robot with P joints and K different states per joint, they can reduce the computational complexity of generating a reachability map from $\mathcal{O}(K^P)$ to $\mathcal{O}(P)$ or even $\mathcal{O}(\log P)$, though they assume that the convolution cost is constant event though it is exponential with respect to K. One additional benefit of using convolution as the fundamental element of the reachability map computation is that we can use the fast Fourier transform algorithm to further accelerate the map generation with small approximation errors [12], [13].

Even though making a significant step towards efficient reachability map generation, most previous works about convolution-based reachability map focus on planar manipulators and 2-dimensional Euclidean motion group $SE(2)$ [9], [10], or some special situations such as ball joint [1], [11] and highly redundant manipulators [14], [15]. This is because the Special Euclidean Group ($SE(D)$) describing the rigid body rotation and translation in D-dimensional Euclidean

*This work was supported by the Natural Science Foundation of China (61725204, 61521002) and Grant from Science and Technology Department of Jiangsu Province, China.

[1] Y. Han, M. Xia, and Y-J Liu are with BNRist, MOE-Key Laboratory of Pervasive Computing, Department of Computer Science and Technology, Tsinghua University, Beijing, China. hyh18@mails. xmf20@mails. and liuyongjin@}tsinghua.edu.cn

[2] J. Pan is with the Department of Computer Science, The University of Hong Kong. jpan@cs.hku.hk

[3] L. Zeng is with the Shenzhen International Graduate School, Tsinghua University. zenglong@sz.tsinghua.edu.cn

[†] Corresponding author

space has $D(D+1)/2$ free parameters, which implies that the computational cost of the convolution over $SE(3)$ would be significantly higher than the convolution over $SE(2)$, given the same resolution K for each dimension. As a result, there exist few works about using convolution techniques to construct the reachability map for typical industrial robots (like KUKA or Universal Robots) working in 3-D workspace.

In this paper, we find that the serial convolution over $SE(3)$ can be replaced by a series of convolutions over $SE(2)$ and 1-dimensional integration. In this way, the computational complexity of generating a 3D robot's reachability map can be reduced to nearly that of a 2D case, which is further accelerated using the fast Fourier transform algorithm and GPU parallel implementation. We use a set of numerical experiments to demonstrate the efficiency of our method, which can reduce the computational time by thousands of times compared to state-of-the-art approaches while generating a map with a similar level of quality. In addition, we use the computed reachability map for the Universal UR5 robot to solve the next-best-view optimization and multi-robot collaborative planning problem for a phenotyping system.

II. Related Work

The concept of the convolution product over the Special Euclidean group is introduced in [9] for numerically calculating the reachability map of discretely actuated manipulators. In particular, for a manipulator consisting of P actuated modules, each with K states, this approach can reduce the computational complexity from $\mathcal{O}(K^P)$ to $\mathcal{O}(P)$, or even $\mathcal{O}(\log P)$. These complexity analysis results, however, hide the timing cost of the convolution calculation in the constant factor, which actually is exponential to the resolution K and is the main bottleneck for efficient map generation.

Some recent works started to apply the convolution technique to the reachability map related robotics problems. Some early methods only consider the reachability maps of planar manipulators with revolutional joints [10], [3]. After that, Dong et al. [1], [11] used reachability maps generated via convolution to perform the inverse kinematics computation first for a robot with discretely actuated ball-joints and then for a robot with active rotation ball-joints. Wang et al. [14] generated the reachability map for a hyper-redundant manipulator by explicitly solving diffusion equations over the motion groups. The convolution technique can also be applied to compute the distribution for joint errors that accumulate from the base to the distal end of kinematic chains, which is a problem closely related to the reachability map generation [16]. By formulating the error propagation as a diffusion process on the motion group (which can not be done for the reachability map), Wang et al. [17] further used a recursive method with second-order accuracy to make the computation feasible for a PUMA industrial robot.

The convolution technique can be accelerated by the fast Fourier transform (FFT). In particular, techniques from the non-commutative harmonic analysis are used to develop fast algorithms for calculating convolution integrals on motion groups [12]. Based on the integral form of the Fourier transform matrix elements, FFT methods are developed for R^3, S^2, and $SO(3)$ and significantly accelerate the convolutions on $SE(3)$. Kim et al. [13] compared the performance of Bayesian filtering on motion groups using the Gaussian approach and the Fourier approach respectively, and demonstrate the advantages of the Fourier approach, especially in situations with large error covariances.

Besides the convolution technique, some other approaches exist for the reachability map computation. For instance, Cao et al. [18] used a Monte Carlo approach to generate the boundary of the 3D workspace region that is reachable for a manipulator. Guan et al. [19] also computed and visualized the reachable workspace for a multi-fingered hand, which is then used for designing manipulation policies.

III. Background

A. Convolution

A general convolution between the scalar functions α and β has the following form:

$$(\alpha * \beta) = \int_{-\infty}^{+\infty} \alpha(\xi)\beta(x-\xi)\, dx. \tag{1}$$

According to [9], the reachability map of a high-DOF manipulator is equal to the convolution product of the lower-dimensional reachabilty map of each DOF on special Euclidean group. Supposing f_1 and f_2 are the reachability maps of a lower segment and an upper segment of the manipulator, the reachability for the composite of two segments will be:

$$(f_1 * f_2)(g) = \int f_1(h) f_2(h^{-1}g)\, dh, \tag{2}$$

where g and h are elements of the special Euclidean group $SE(D)$, which is a $D(D+1)/2$ dimensional Lie group describing translation and rotation in the D-dimensional Euclidean space. Each element of $SE(D)$ can be parameterized as:

$$\begin{pmatrix} R & \overline{v} \\ \overline{0}^T & 1 \end{pmatrix}, \tag{3}$$

where $R \in SO(D)$ is a $D \times D$ matrix representing the rotation on the group and $\overline{v}$ is a $D \times 1$ vector representing the translation.

Given a robot arm with N different segments, we can generate the reachability map f_i for the i-th segment, and then convolve them together to get the final reachability map for the entire robot:

$$f_{1 \ldots N} = (f_1 * f_2 \ldots * f_N). \tag{4}$$

We use a voxel map to represent the reachabilty map and to perform the convolution operation where each dimension of $SE(D)$ is discretized into segments with equal sizes. In particular, for $SE(2)$, we will divide the x, y and θ axes into N_1, N_2 and N_3 segments, respectively and the total number of voxel cells is $N_1 \times N_2 \times N_3$. In this way, the convolution

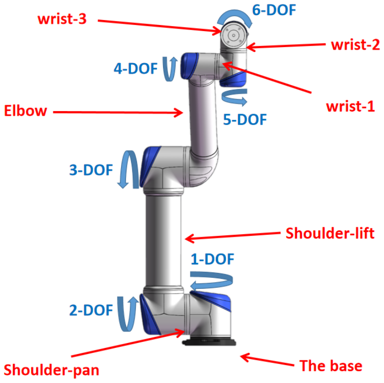


Fig. 2. We denote these 6 DOFs and 7 links of the UR manipulator. The description of each link of the UR manipulator in the following text is consistent with this figure.

on $SE(2)$ can be implemented as:

$$\begin{aligned}
&(f_1 * f_2)(x, y, \theta) \\
=&\sum_{l=0}^{N_1} \sum_{m=0}^{N_2} \sum_{n=0}^{N_3} f_1(x_1, y_1, \theta_1) \\
&\cdot f_2\Big((x - x_1)\cos\theta_1 + (y - y_1)\sin\theta_1, \\
&\quad -(x - x_1)\sin\theta_1 + (y - y_1)\cos\theta_1, \\
&\quad (\theta - \theta_1)\text{mod}2\pi\Big)\, \Delta x\, \Delta y\, \Delta\theta.
\end{aligned} \tag{5}$$

B. Comparison of Convolution on $SE(3)$ and $SE(2)$

Convolution on $SE(3)$ is much more complicated than that on $SE(2)$. First, because $SE(2)$ is a 3D space while $SE(3)$ is 6D, the computational complexity of $SE(3)$ is at least squared of $SE(2)$. In addition, $SE(3)$'s topology also makes the convolution more challenging. In particular,

$$SE(3) \cong SO(3) \rtimes R^3 \tag{6}$$

and thus $SE(3)$ is not a Cartesian product but a semi-direct product of $SO(3)$ and $\mathbb{R}^3$. To implement this semi-direct product, we have to appropriately parameterize $SO(3)$. Unlike $SO(2)$ can be represented easily as a rotation angle, $SO(3)$ consists of three rotations around three axes and it is a difficult parameterization approach that Euler angles cannot well characterize $SO(3)$'s topological structures, which results in the difficulty of discretizing $SO(3)$ finely and uniformly and also the convolution on $SE(3)$.

Due to aforementioned reasons, existing approaches are limited to planar robots or tasks on $SE(2)$ or some special subset of $SE(3)$. Thus there exists few methods using convolution to compute the reachability map for a high-DOF manipulator like KUKA or Universal robotic arms. In this paper, we can bypass the difficulty of $SE(3)$ convolution when computing the reachability map for a many high-DOF manipulator by leveraging the fact that the calculation can be achieved alternatively by only using the $SE(2)$ convolution.

IV. Method

A. Interplanar Integration of Intra-planar Convolution (I^3C)

We propose a general approach that reduces the computational complexity of reachability map computation and is applicable to general high-DOF manipulators. In the following part, we will use the Universal UR robot as one example to describe our method.

Our method consists of three steps. First, we divide the manipulator joints into several groups where links in each group move in the same plane. In particular, we find all joints that are always coplanar and group them together. And each of the remaining isolated joints is also treated as an individual group. Second, for joints in the same group, we perform convolution on $SE(2)$ according to Equation 5, because their links are restricted to move in the corresponding plane. Finally, we perform convolution between adjacent planes to obtain the final reachability map. Thus, we name our new solution the Interplanar Integration of Intra-planar Convolution (I^3C).

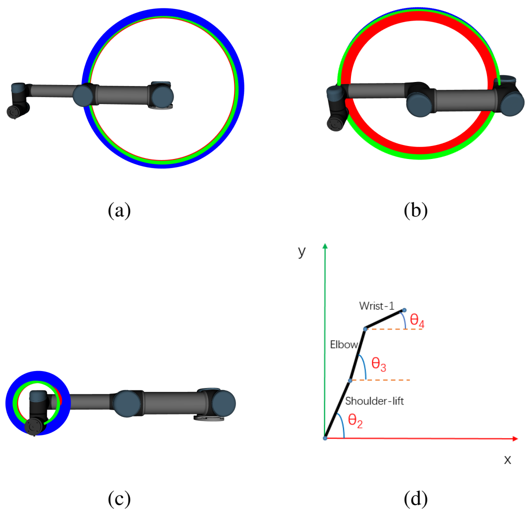


Fig. 3. Illustration for the intra-planar convolution: (a-c): the local reachability map of each link in the group of shoulder-lift, elbow and wrist-1. (d): the coplanar links in $x - y$ plane.

1) Intra-planar Convolution: According to I^3C, the UR robot arm can be divided into three groups.

As shown in Figure 2 and Figure 3, the Shoulder-lift, Elbow, Wrist-1 of the UR manipulator will always move on the same plane and thus their convolution can be fully expressed on $SE(2)$. We can get the convolution result f_{23} of Shoulder-lift and Elbow directly by Equation 5. The subscript of f represents the indices of joints involving in the convolution. In the same way, f_{23} and f_4, the reachability map of the wrist-2, can be convolved together using Equation 5 to get f_{234}.

All the other joints, including the shoulder-pan and wrist-2, are not moving in the same plane and thus a single-element group will be made for each of them.

2) Interplanar Integration: In this step, we need to perform the convolution between the reachability maps between adjacent groups.

For UR robotic arm, the plane of Wrist-2 and the plane of Shoulder-lift, Elbow and Wrist-1 are vertical to each other as shown in Figure 4, and the reachability map for the union of these groups can be computed as:

$$f_{2345}(x, y, z, \theta, \varphi) = \chi_{\{L \sin \varphi' = z\}} f_{234}(x', y', \theta'), \tag{7}$$

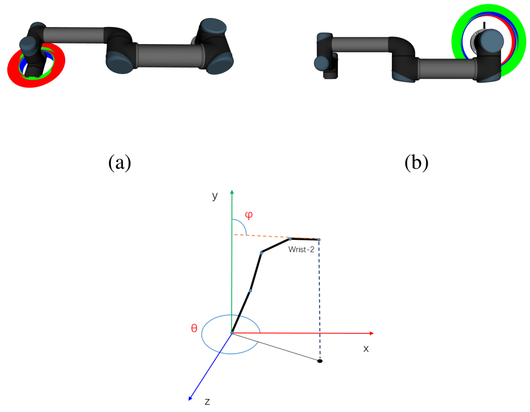
where

$$\chi_{\{L \sin \varphi' = z\}} = \begin{cases} 1, & \text{if} \quad L \sin \varphi' = z \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

and

$$\begin{aligned} \theta' &= -\arctan(\tan \varphi \sin \theta) \\ \varphi' &= \arcsin(\sin \varphi \cos \theta) \\ x' &= x - L \sin \varphi \sin \theta \\ y' &= y - L \cos \varphi. \end{aligned} \tag{9}$$

Here φ' represents the angle of the wrist-2 joint. Although we have added an angle axis φ and a coordinate axis z for representing the reachability map on $SE(3)$, the computational complexity is even lower than the convolution on $SE(2)$.


(a)

(b)

(c)

Fig. 4. Illustration for the inter-planar convolution: (a): the local reachability map of Wrist-2. (b): the local reachability map of Shoulder-pan. (c): the convolution in $x - y - z$ coordinates with the illustration of θ and ϕ used in Equation 7 and Equation 10.

Next, we model the integration towards the base link to compute the entire reachability map, in which the rotation of the Shoulder-pan must be taken into account:

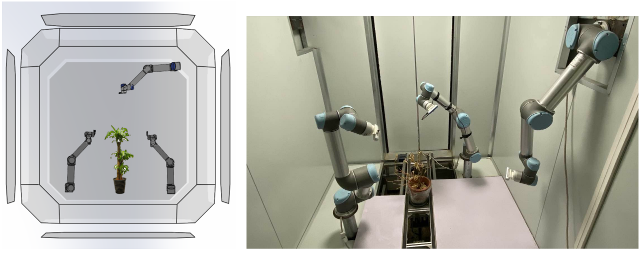
$$f_{12345}(x, y, z, \theta, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} f_{2345}(x, \\ y \cos \psi' + z \sin \psi', z \cos \psi' - y \sin \psi', \theta - \psi', \varphi)\, d\psi' \tag{10}$$

where ψ represents the joint angle of Shoulder-pan.

B. Applications for Reachability Map

To facilitate understanding, we start from a real application scenario to illustrate how to apply a reachability map generated by I^3C. We apply the reachability map in an updated multi-manipulator plant phenotyping system [20] for improving the efficiency of motion planning. This system was designed to provide fast, precise, and non-invasive measurements for robot-assisted high-throughput plant phenotyping. To achieve this goal, this system was equipped with three UR5 robotic arms that can be moved simultaneously in each round of phenotyping data acquisition, using the depth camera (Intel RealSense D435) mounted at each robotic arm. Under ideal conditions, the cycle of this system is shooting-image processing and modeling-motion planning-selecting the next best viewpoint. Repeat this cycle until a high-quality reconstruction model is obtained.

It is challenging to implement this ideal cycle. The next best viewpoint selected usually fail to plan, and it is often necessary to repeatedly select viewpoints, which makes the real-time performance of the system unable to ensure. The solution currently used is to pre-select a large number of reachable viewpoints and then calculate the best from these viewpoints' combinations. This scheme discretizes the infinite continuous space, limiting the choice of the best viewpoint. To ensure both the real-time performance and continuous viewpoint space, we introduce our reachability map into the system.


(a) Rendered system design

(b) A snapshot of real system

Fig. 5. An updated multi-robot plant phenotyping system [20].

1) base placement: In this scenario, the perspective coverage should be guaranteed while the best location cannot be found solely from the coverage rate. It is also necessary to consider the factor of high reachability by setting a threshold on reachability map. To achieve both higher coverage and higher reachability, we try to find the optimal base placement. For each manipulator, we discrete all its possible placement spaces. Suppose that each manipulator has N possible base positions, there are N^3 kinds of all possible combinations. To simply calculate the coverage rate, we randomly sample three-dimensional orientations. The calculation process is as Algorithm 1.

2) sample by the reachability map: Spending a lot of time repeatedly selecting the best viewpoint and trying to plan is the reason why real-time preformance cannot be achieved. In other words, if the selected viewpoint has a higher planning success rate, the real-time performance will be relatively guaranteed. Therefore we apply the reachability map by selecting the best viewpoint from high-reachability voxels. We set a threshold to divide voxels into sampleable and non-sampleable part, and the best viewpoint will only be selected from the sampleable voxels.

Algorithm 1 Coverage rate$(A, B, M, O, threshold)$

```
$A$: $N$ three-dimensional orientations
$B$: 6D position of three manipulators
$M$: Reachability map
$O$: Shooting origin
Find all reserved voxels by $B, M, threshold$
Express ten thousand rays by $O$ and $A$
$n$=the number of rays passing through reserved voxels
Result=$n/N$
```

However, the pure map does not take into account the continuity of motion. In addition to the reachability, the planning success rate is also related to the start and target position. The start position (home position) is shown in Figure 5(b), calculated by finding the mean joint value of random samples without collision. To express reachability more objectively, the reachability map M should be adjusted from the home state:

$$M' = M/D^k \tag{11}$$

where D represents the Euclidean distance between voxels and home state voxel. k is a constant which is used to adjust the reachability and movement continuity.

V. Experiments

A. Time Efficiency

As aforementioned in Section I, convolution calculations can be accelerated by deploying on GPUs, as well as applying FFT algorithms. Compared to brute force method (FK) and traditional convolution on $SE(3)$, it is advantageous to get reachability maps through our I^3C.

All experiments about time efficiency are deployed on a server with Intel E5-2600-v3 CPU, 256 GB RAM, and NVIDIA RTX2080Ti. To control the variables, we only used a single GPU and CPU. The experimental data is as Table I.

TABLE I

Time Efficiency Comparison of Intra-planar Convolution. The impact factors taken into consideration are: CPU, GPU, Interplanar Integration of Intra-planar Convolution (I^3C), FFT. The data units in the table are seconds and $2^n, 2^m$ determines the data size (2^n for position, 2^m for orientation). The reachability map will be divided into $2^n \times 2^n \times 2^m$ voxels for I^3C in SE(2), while it will square without I^3C on $SE(3)$. / indicates exceeding memory space.

method	$2^7, 2^5$	$2^7, 2^6$	$2^8, 2^5$	$2^8, 2^6$	$2^8, 2^7$	$2^9, 2^6$
CPU	>500000	/	/	/	/	/
GPU	/	/	/	/	/	/
CPU+I^3C	29.5	117.89	469.65	1856.31	7516.45	29699.38
GPU+I^3C	1.06	4.10	16.35	67.07	273.52	1099.89
CPU+FFT+I^3C	0.34	0.649	1.75	4.89	13.58	30.26

The fast Fourier transform (FFT) has been proposed for $SE(3)$ in previous work, though most of them only provide theoretical descriptions with limited details about how to implement $SE(3)$ FFT in practice. By I^3C, we avoid the difficulty of $SE(3)$ FFT implementation and only need to focus on $SE(2)$ FFT. The computation of convolution integrals on $SE(2)$ has been modeled by G. S. Chirikjian[12].

Usually, the convolution kernel is relatively small (such as deep learning), and the FFT has almost no acceleration effect. But for our scenario, the two convolved reachability maps are almost of the same size, which means that the acceleration effect of FFT can be better utilized. Assuming the size of both sides of the convolution are N (total number of voxels), the complexity of the convolution calculation will be $\mathcal{O}(N^2)$. After the introduction of FFT, the complexity will be reduced to $\mathcal{O}(N \log N)$.

TABLE II

Time Efficiency Comparison of Interplanar Integration.

method	$2^7, 2^5$	$2^7, 2^6$	$2^8, 2^5$	$2^8, 2^6$	$2^8, 2^7$	$2^9, 2^6$
CPU	>500000	/	/	/	/	/
GPU	/	/	/	/	/	/
CPU+I^3C	42.83	174.89	701.96	2464.12	10225.60	42369.43
GPU+I^3C	1.26	5.27	21.08	84.97	340.96	1370.38

FFT cannot be applied to Interplanar Integration. From Table I and Table II, we can draw three conclusions. Firstly, I^3C greatly reduces both time and space complexity. Secondly, if we apply FFT to Intra-planar Convolution, the Interplanar Integration will become the time bottleneck. Thirdly, GPU parallel and FFT acceleration can significantly speed up the calculation time.

To prove the accuracy of I^3C, we have to find a benchmark for comparison. However, the brute force method cannot guarantee the accuracy in the case of high DOF and the convolution on $SE(3)$ for manipulator even cannot be modeled directly. In desperation, we compare I^3C with the brute force method.

TABLE III

From the convolution result of two joints to all joints. The subscript represents the joint of the convolution.

	f_{23}	f_{234}	f_{2345}	f_{12345}
Similarity	0.9943	0.9876	0.8661	0.8402

We uniformly sample the value of each joint and obtain the end poses of the manipulator through FK. Then voxelize these end poses to obtain data in the same form as the convolved reachability map. Calculate the mean reachability value of the two maps, and then select all voxels with a reachability higher than the mean reachability value. The similarity is defined as the proportion of these voxels overlap. We believe that the decrease in similarity of f_{2345} and f_{12345} is mainly caused by the growth of voxel dimensions, distortion of brute force method and discretization error.

B. Test in Application Scenario

As aforementioned in Section IV.B, we use a reachability map in base placement and viewpoint collection to ensure both the real-time performance and continuous viewpoint space. Our reachability map can express the reachability of a certain end pose when the base's position and orientation are determined.

All experiments in this part are deployed on the same server in V-A and ROS with the open motion planning library (OMPL) [21].

We use a 6D reachability map with a size of $10^3 \times 8^3$ (10 for position, 8 for orientation) for each manipulator, and the total number of voxels is $512,000$. We select $k = 1, 1.5, 2$ in the experiment. Since different k will lead to a different reachability of M', we set a dynamic threshold expressed by $n \times p$.

1) Base placement: We discrete all its possible placement spaces in units of $10\,\mathrm{cm}$ in a plane for each manipulator. Suppose that each manipulator has N possible base positions, there are N^3 kinds of all possible combinations. To find the best base position, N^3 coverage rate calculation should be done. For each k and n, the calculation of Algorithm 1 has to be repeated. The best coverage rate of each k and n are shown in Table IV.

TABLE IV

COVERAGE RATE FOR EACH k, p AND n. WHILE FINDING THE BEST BASE POSITION, IT ALSO FOUND THE BEST COVERAGE RATE.

	$n=0$	$n=1$	$n=2$	$n=3$	$n=5$
$k=2, p=1$	1	0.919	0.648	0.500	0.363
$k=1.5, p=3$	1	0.990	0.914	0.799	0.551
$k=1, p=10$	1	1	0.993	0.956	0.859

2) Sample by the reachability map: As usual, the higher the reachability, the higher the success rate of path planning. For the multi-manipulator plant phenotyping system mentioned in Wu [20], the point of view only needs to be selected in part of the area instead of the entire workspace (Wu [20] is only selected on a ball). In this case, we set a threshold for filtering voxels from the reachability map and then only collect viewpoints from the reserved voxels. Collect viewpoints and get the joint value of the manipulator corresponding to each viewpoint by inverse kinematics. Repeat this sampling process until the resulting three sets of joint values pass the collision detection. Finally, calculate the planning success rate, which is shown in Table V.

TABLE V

PLANNING SUCCESS RATE. THE DYNAMIC REACHABILITY THRESHOLD WAS SET TO $n \times p$. THE VOXELS IN M' WILL BE SET TO 0 IF THE REACHABILITY IS LOWER THAN $n \times p$. FOR EACH RANDOM SAMPLING PROCESS, WE FIRST RANDOMLY SAMPLE A VOXEL ACCORDING TO THE REACHABILITY, AND THEN RANDOMLY SAMPLE A 6D POSE IN THE SPACE OF THE VOXEL. FINALLY, CALCULATE THE PLANNING SUCCESS RATE OF $1,000$ COLLISION-FREE SAMPLES.

	$n=0$	$n=1$	$n=2$	$n=3$	$n=5$
$k=2, p=1$	0.646	0.699	0.718	0.729	0.760
$k=1.5, p=3$	0.604	0.654	0.667	0.671	0.714
$k=1, p=10$	0.596	0.601	0.624	0.659	0.679

The planning success rate of the random sample is about 0.55. Briefly summarize Table V, we can firstly find that the success rate will be higher when the k becomes larger, which is in line with the law (the closer the voxel, the easier it is to plan). Secondly, a high dynamic reachability threshold will lead to a high planning success rate.

To make the threshold work better, the selection basis of the p value: when $n = 1$, more than half of the voxels are screened out. A higher k value and a higher threshold setting can improve the planning success rate, but the coverage rate will decrease. There needs to be a trade-off between k, n, and p. According to our experience, when the coverage rate is above 0.9, only some low-value viewpoints at the bottom cannot be covered. So we set $k = 2$, $p = 1$, $n = 1$, and the success rate is still 15% higher than the case without reachability map. The optimal base position is shown in Figure 6.

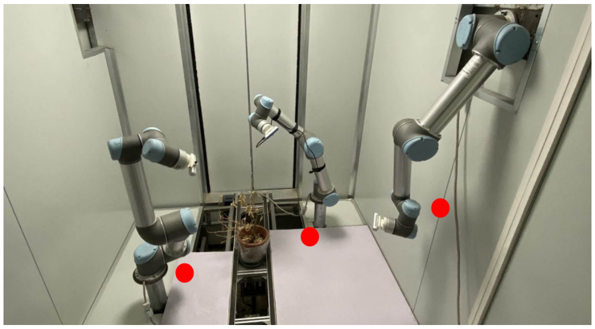


Fig. 6. The red dot indicates the best position of the base.

3) Verify the new base: To verify the effectiveness of the new base in the Next Best View (NBV) of our multi-system, we find 10,000 viewpoints by NBV and do inverse dynamics on three UR5.

TABLE VI

COMPARISON OF THE EFFECTIVENESS OF THE ORIGINAL AND NEW BASE. 0,1,2,3 REPRESENT THE NUMBER OF UR5 THAT SUCCESSFULLY SOLVED INVERSE DYNAMICS.

	0	1	2	3
Original base	3.35%	80.88%	15.72%	0.05%
New base	0.89%	12.3%	25.28%	61.53%

After changing the base, there is no significant increase in the motion planning success rate for the specific end poses of three UR5. However, for more than 60% of viewpoints, all the UR5 manipulators can obtain inverse kinematics solutions, which means more groups of end pose will correspond to a groups of viewpoints (3 viewpoints) and we can try multiple sets of end poses to get a plan for a groups of viewpoints. For a group of viewpoints given by NBV, the planning success rate of the new base can reach 60%, while the original base is only 28%.

VI. Conclusions

In this paper, we propose I^3C to reduce the computational complexity of obtaining reachability maps for high-DOF manipulators in 3D workspace. GPU parallel computing and Fast Fourier transform are also leveraged to further accelerate the computation procedure. Experiments show that we have greatly improved computational efficiency and still keep a high quality. A real application of the reachability map in a multi-robot system using three UR5 robotic arms is presented to show the high practical value. With the help of reachability map, the planning success rate of our multi-robot system has increased by 15% and a best base placement has also been found to further improve the system. For a group of viewpoints, the planning success rate of the best base has increased by 32%.

References

[1] H. Dong, T. Fan, Z. Du, and G. S. Chirikjian, "Inverse kinematics of active rotation ball joint manipulators using workspaces density functions," in *Advances in Reconfigurable Mechanisms and Robots II*. Springer, 2016, pp. 633–644.

[2] O. Porges, T. Stouraitis, C. Borst, and M. A. Roa, "Reachability and capability analysis for manipulation tasks," in *ROBOT2013: First Iberian robotics conference*. Springer, 2014, pp. 703–718.

[3] H. Dong and Z. Du, "Obstacle avoidance path planning of planar redundant manipulators using workspace density," *International Journal of Advanced Robotic Systems*, vol. 12, no. 2, p. 9, 2015.

[4] K. Maeda and E. Konaka, "Model predictive control method for position control and obstacle avoidance of hyper-redundant binary manipulator," in *2014 Proceedings of the SICE Annual Conference (SICE)*. IEEE, 2014, pp. 1262–1267.

[5] M. Wagner, P. Heß, S. Reitelshöfer, and J. Franke, "Reachability analysis for cooperative processing with industrial robots," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–6.

[6] O. Porges, R. Lampariello, J. Artigas, A. Wedler, C. Borst, and M. A. Roa, "Reachability and dexterity: Analysis and applications for space robotics," in *Workshop on Advanced Space Technologies for Robotics and Automation-ASTRA*, 2015.

[7] J. Dong and J. C. Trinkle, "Orientation-based reachability map for robot base placement," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 1488–1493.

[8] N. Vahrenkamp, T. Asfour, and R. Dillmann, "Robot placement based on reachability inversion," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1970–1975.

[9] G. S. Chirikjian and I. Ebert-Uphoff, "Numerical convolution on the euclidean group with applications to workspace generation," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 123–136, 1998.

[10] H. Dong, Z. Du, and G. S. Chirikjian, "Workspace density and inverse kinematics for planar serial revolute manipulators," *Mechanism and Machine Theory*, vol. 70, pp. 508–522, 2013.

[11] H. Dong, T. Fan, Z. Du, and G. Chirikjian, "Inverse kinematics of discretely actuated ball-joint manipulators using workspace density," in *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection, 2015.

[12] A. B. Kyatkin and G. S. Chirikjian, "Algorithms for fast convolutions on motion groups," *Applied and Computational Harmonic Analysis*, vol. 9, no. 2, pp. 220–241, 2000.

[13] J. S. Kim and G. S. Chirikjian, "Bayesian filtering for orientational distributions: A fourier approach," in *2015 18th International Conference on Information Fusion (Fusion)*. IEEE, 2015, pp. 748–753.

[14] Y. Wang and G. S. Chirikjian, "Workspace generation of hyper-redundant manipulators as a diffusion process on se (n)," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 399–408, 2004.

[15] R. J. Murphy, M. S. Moses, M. D. Kutzer, G. S. Chirikjian, and M. Armand, "Constrained workspace generation for snake-like manipulators with applications to minimally invasive surgery," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5341–5347.

[16] Y. Wang and G. S. Chirikjian, "Error propagation on the euclidean group with applications to manipulator kinematics," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 591–602, 2006.

[17] ——, "Nonparametric second-order theory of error propagation on motion groups," *The International journal of robotics research*, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[18] Y. Cao, H. Zang, L. Wu, and T. Lu, "An engineering oriented method for the three dimensional workspace generation of robot manipulator," *Journal of Information & Computational Science*, vol. 8, no. 1, pp. 51–61, 2011.

[19] Y. Guan, H. Zhang, X. Zhang, and Z. Guan, "Workspace generation for multifingered manipulation," *Advanced Robotics*, vol. 25, no. 18, pp. 2293–2317, 2011.

[20] C. Wu, R. Zeng, J. Pan, C. C. Wang, and Y.-J. Liu, "Plant phenotyping by deep-learning based planner for multi-robots," *IEEE Robotics and Automation Letters (to be presented at IROS 2019)*, vol. 4, no. 4, pp. 3113–3120, 2019.

[21] I. A. Sucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.