

Cellular Decomposition for Non-repetitive Coverage Task Ensuring Least Discontinuities

Tong Yang¹, Jaime Valls Miro², Qianen Lai¹, Yue Wang^{1*} and Rong Xiong¹

Abstract—There are many non-repetitive coverage tasks which use a high-dimensional path to cover low-dimensional space, such as the polishing task using the manipulators. Due to the non-bijective mapping between the workspace and the joint-space, a continuous coverage path in the workspace is truncated in the joint-space, and there are multiple choices of configurations to cover a same waypoint of the coverage path. The discontinuous point implies a suspension of the coverage task, which is difficult to handle. For example, the suspension of the polishing task means the lift-off the end-effector from the surface of the object, where the transition between the force/position control is required, which is more complicated than the ordinary coverage process.

In this paper, motivated by the non-repetitive coverage task of non-redundant manipulators, we first prove that the least number of discontinuity is a parameter of the environment setting, independent to the choice of coverage paths, thus has a minimum. Then, a cellular decomposition method generating all optimal cellular decompositions is proposed. The algorithm is operational in any dimension, and we illustrate it through polishing tasks. Through simulated experiments, the least number of discontinuities is shown as a novel criterion of the quality of the placement of the manipulator (or the object). Besides, the method helps to solve the problem of multiple inverse kinematic (IK) solutions. In real-world experiment, the physical coverage path is generated to show the applicability of the proposed algorithm.

Index Terms—cellular decomposition, coverage task, non-redundant manipulator

I. INTRODUCTION

THE non-repetitive coverage task of a given object is an important application of the manipulators, which is a kind of coverage path planning (CPP) problem. For example, the polishing task requires a coverage path of the end-effector (EE) to pass all points on the surface of a given object for exactly one time, with the orientation of the EE perpendicular to the surface. Typically, the dimension of the joint-space is higher than the workspace, such as using a non-redundant manipulator to polish a surface, and the inverse kinematic mapping between the joint-space and the workspace is non-bijective, which causes two difficulties in coverage task. One is that planning in the joint-space cannot ensure non-repetitive visiting, since the mapping is not injective, so the coverage

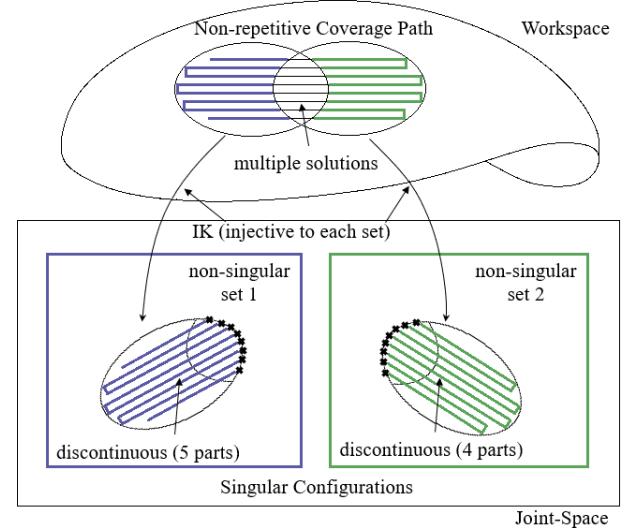


Fig. 1. A diagram showing the relation between the joint-space and the workspace. The non-singular configurations form disjoint sets which are denoted by different color. The colored parts of the path have unique IK solutions. However, for the path with black line, there are multiple IK solutions, which requires further decision. Besides, the continuous path in the workspace is no longer continuous after mapped into the joint-space. In this example, whatever choice among the multiple IK solutions, 8 discontinuities are required, since the path has five separated parts in set 1 and four parts in set 2.

path must be designed directly in the workspace [1]. see Figure 1. The other problem is that a continuous coverage path in the workspace may be truncated into many pieces after mapped into the joint-space, where the discontinuities are difficult to handle. For example, the discontinuity in polishing task means lifting the EE off the surface of the object, adjusting the pose of the manipulator and re-touching the surface. To handle such process, the transition between the position control and the force control is required [2] [3] [4], which is more complicated than the usual movement along the surface.

People have noticed that the singularities are the cause of the bifurcation of the joint-space [5] [6]. In the coverage task, the singular configurations are useless because of the loss of manipulability. Thus, all non-singular configurations form disjoint sets in the joint-space. See Figure 1 for illustration. What we further notice is that the IK mapping from the reachable points in the workspace to a single set of configurations is injective. The reason is that, the different IK solutions come from different solutions of the inverse trigonometric function, between which the joint angle goes through a singular angle. As a result, different IK solutions for a same pose belongs to different sets. Due to the constraints,

¹ Tong Yang, Qianen Lai, Yue Wang and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, P.R. China.

² Jaime Valls Miro is with the Centre for Autonomous Systems (CAS), Faculty of Engineering, University of Technology Sydney (UTS), NSW 2007 Sydney, Australia.

* Corresponding Author.

E-mail address: wangyue@iipc.zju.edu.cn

commonly the whole workspace cannot be mapped into a single set, then the continuous joint-space path must visit some singularities, inevitably causing lift-offs.

There are many significant works looking for the optimal coverage paths [7] [8], but the variation of the cost between different continuous coverage paths is far less than that of one extra discontinuity. Also, there are many literatures [2] [3] [4] [9], [10] discussing reducing the cost during the contacting and maintaining smooth and stable transition, none of them notices that a coverage path without unnecessary lift-offs is solvable. [11] considers optimizing the position of the mobile manipulator for coverage task, which strengthens the requirement of a valid criterion of the relative pose between the manipulator and the object.

In this paper, we make the following contributions:

(1) Prove that the least number of lift-offs for a non-repetitive coverage task using non-redundant manipulator is a parameter of the environment setting (the related pose between the manipulator, the object and all obstacles), independent to the choice of the physical coverage path. Hence it becomes a criterion evaluating the quality of the placement of the manipulator (or the object).

(2) Propose a novel cellular decomposition method which divides the surface into least number of cells, ensuring that each cell can be covered without lift-off through the specified configurations.

The remainder of this paper is organized as follows. Section II reviews the existing literature. Section III states the problem, models the problem into an abstract form that painting a graph using least number of colors and proves that the number of discontinuities is independent to the coverage path. Section IV discusses the process of solving a single element. Section V presents the iterative strategy to solve the whole problem. Section VI shows the experimental results, and Section VII concludes the paper. In the sequel, we use “optimal path” to represent a coverage path that requires least number of discontinuities, and present our algorithm in the language of the polishing task.

II. RELATED WORK

The CPP problem is looking for a path that passes all points within a given area. Since it can be formulated into the Travelling Salesman Problem (TSP) [12] [13], which is thought as an NP problem, almost all state-of-the-art methods first decompose the given area and then solve the CPP problem in each cell.

The CPP algorithms can be mainly divided into two categories: the exact cellular decomposition methods and the morse-based cellular decomposition methods. The exact cellular decomposition methods [14] divide the free space into several simple, easy sub-regions, and use conventional coverage paths, such as the trapezoidal path [15] or the boustrophedon path [16] [17], to finish the coverage in each cell. The morse-based cellular decomposition methods apply the divisions of the free space based on the critical points of Morse functions [18] [19] to present more flexible shape for cells than the exact cellular decomposition.

The optimal CPP algorithms mainly focus on the path length and time to completion. Atkar et al. presented a CPP algorithm for the spray painting robots [7]. It maps the conventional coverage path to the 2.5D surface and optimizes the coverage path through choosing optimal starting point. Huang presented an optimal line-sweep based method for cellular decomposition algorithms in planar spaces [8]. It covers each cell with sweeping path going in different directions to minimize the number of returns. However, this approach does not take into account the cost of traveling between cells. Jimenez proposed to use a genetic algorithm to achieve optimal coverage [20]. The free space is divided into sub-regions using the trapezoidal cellular decomposition, then a genetic algorithm is used to plan an optimal path that covers all the subregions. Mannadiar and Rekleitis proposed an algorithm that achieves complete coverage of spaces while minimizing the path [21]. The algorithm encodes the cells to be covered as edges of the Reeb graph. Then, the optimal solution to the Chinese Postman Problem is used to calculate an Euler tour, which guarantees complete coverage and the minimum of path length.

Searching for such a path with least number of discontinuities for the manipulator is not trivial. However, none of the existing literatures paid attention to it. Since the mobile robots cannot leave the ground, it is impossible for a CPP algorithm designing for the mobile robots that considers this contact loss. As for the contact task of the manipulator, most of the literatures [9] [10] focus on the control strategies instead of avoiding detaching and re-touching the surface as many as possible.

III. PROBLEM FORMULATION

In this section, we first state the optimal cellular decomposition problem of least number of discontinuities. Then, the equivalent problem that painting a graph into least number of pieces is created. Finally, we show that the least number of discontinuities is independent to the choice of the physical coverage path.

A. Problem Statement

Given the surface of an object, the structure of the manipulator, the shape of other obstacles and their relative poses, the required coverage path consists of all valid configurations which satisfy the following constraints:

(1) Kinematic Constraint: The manipulator is collision-free during the whole process. And when the EE contacts the surface, its z -axis is align with the normal vector of the surface at the contact point.

(2) Force Constraint: When the EE contacts the surface, the manipulator should be able to exert the required force on the EE in the z -axis.

(3) Manipulability Constraint: When the EE contacts the surface, the manipulator should keep well-conditioned (under some given manipulability measure) to deal with the perturbation.

One lift-off of the manipulator is denoted by a continuous piece of configurations in the path during which the EE does

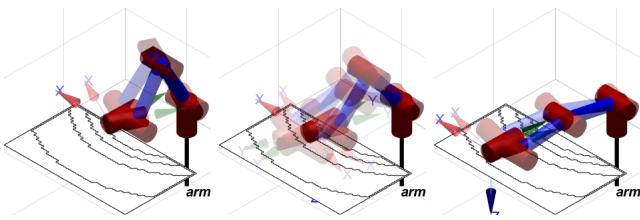


Fig. 2. The manipulator is polishing a planar rectangular object. The nearest and the farthest area are unreachable. Within the middle area, there are four kinds of configurations to polish the middle part, but only two of them can also polish the inner part and the outer part. Note that the vivid configurations in figures can be continuously reached (without lifting off the EE).

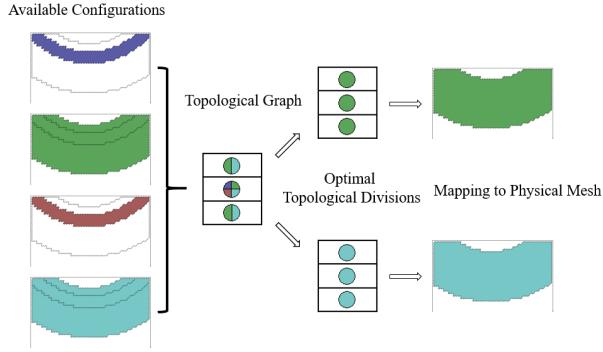


Fig. 3. Flowchart of our algorithm in solving the problem of Figure 2. All configurations are divided into four disjoint sets, represented by different colors. The small circle filled in with color(s) shows all possible colors to paint the corresponding area. The topological graph is created based on the distribution of the colors and being solved. Finally, in this example, we can get two optimal options which both require zero lift-off.

not contact the surface. Then, the optimal cellular decomposition problem is to find a valid path of configurations which covers the workspace non-repetitively and ensures the minimum number of the lift-offs. In this problem, the contact area between EE and the surface is seen as a particle.

B. Problem Modeling

Without loss of generality, the input data is a triangular mesh, with all vertices and edges fully known. Then the normal of each vertex and all valid IK solutions to cover it are also known. Since the manipulator is non-redundant, the number of IK solutions for each vertex is finite. Although the mesh is a discretized data structure, we say two vertices are “continuous”, if there is a list of edges of the mesh connecting them. And we say two configurations are “continuous” for the manipulator to reach, if the vertices are continuous, and the distance of these two configurations in the joint-space is near enough. The distance threshold for the continuity is easy to judge, since the manipulability constraint abandons the configurations which are close to the singularities, thus creates an apparent gap between disjoint sets. Typically, just the signs of the joint angles are enough to judge the continuity of two configurations. For example, it is easy to see that three vivid configurations in Figure 2 are continuous, and the greyed out configurations in a same figure are discontinuous pairwise.

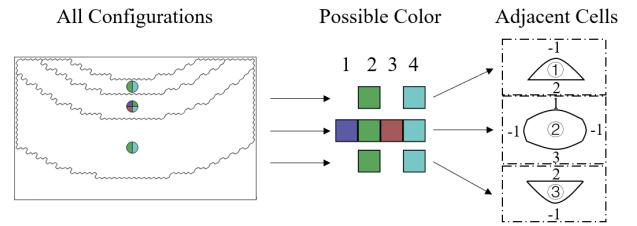


Fig. 4. The elements of a cell. All colors are indexed. Each cell corresponds to a connected area on the surface of the object and has a list of possible colors. The cell stores the index of its adjacent cell in order. The index of the unreachable area is denoted by -1 . We use curves to show that the edge is topological but not physical.

First, we assign the color of all configurations. Starting from an unassigned configuration, using a floodfill-like algorithm on the mesh, all configurations which are continuous with the chosen one are easy to find and are assigned with a same number, which is the index of the corresponding color. After repeating the assignment process, all configurations have a color, and the continuous configurations have a same color. Note that the IK solutions for the same vertex must have different colors. So it is suitable to show the distribution of a color through drawing them on the mesh. For example, Figure 3 shows the distribution of four colors of the example in Figure 2.

Second, we divide the mesh into several “cells”, with each vertex belonging to a cell. The cell is a connected open region containing all vertices which are continuous and can be covered through same kinds of colors. For the generation of a cell, we pick up an unassigned vertex, using a floodfill-like algorithm to find all continuous vertices which have same kinds of colors. These vertices belongs to a single cell. In Figure 3, we draw a small circle filled in with multiple colors to show the possible colors for a cell. After repeating the process, all vertices are assigned.

Third, after the creation of all cells, we say there is a topological edge between two cells, if there is a physical edge of the mesh which connects two vertices from different cells. We may see the intersection of topological edges in the following figures, but note that they are formally drawn, since each edge connects only two vertices which are impossible to belong to three or more different cells. In all, the structure of the topological graph is uniquely decided by the mesh.

Finally, the cell records the possible colors for covering the points and the index of its adjacent cells in order, see Figure 4. Note that in practical application, since the area of the surface is finite, and each cell must have least size to keep the decomposition meaningful, the number of cells must be finite.

After creating the topological graph, the original problem is equivalent to painting all points with one of their available colors, ensuring the minimum pieces of colors. And for a fully-filled graph, the color of a point uniquely specifies one of the valid IK solutions to cover it.

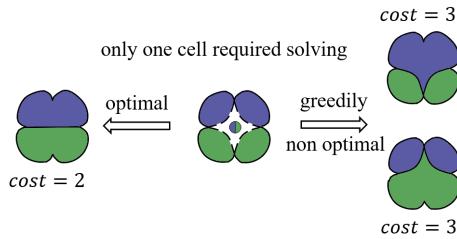


Fig. 5. In this graph, the boundary four cells only have one possible color and only the middle one needs solving. If the colors are chosen greedily, the middle cell must be filled in with pure color (like the right side), whose final cost will be 3. However, if the middle cell is cut into two parts, then two pairs of adjacent cells are connected and the cost will be 2, which reaches the optimality. Note that the cutting path is arbitrary with the same function.

C. Independence from the physical coverage path

Since the manipulator is omni-directional in the joint-space, we can always find a continuous coverage path for a connected region in the joint-space. From the above definition of the cell, all points in a same cell have same kinds of colors, thus can be covered through some kinds of configurations which are represented by the colors, without lifting off the EE. Let the number of the cells be n , then if we design a coverage path within each cell, and directly concatenate these paths through paths with the EE leaving the surface, the total number of lift-offs is at most $n - 1$. Since the number of cells is finite, the number of lift-offs is also finite. And because the creation of the initial graph depends only on the IK solutions of all points but not their order, the minimum number of lift-offs is a parameter of the environment setting and independent to the choice of the physical coverage path.

IV. ENUMERATIVE SOLVER

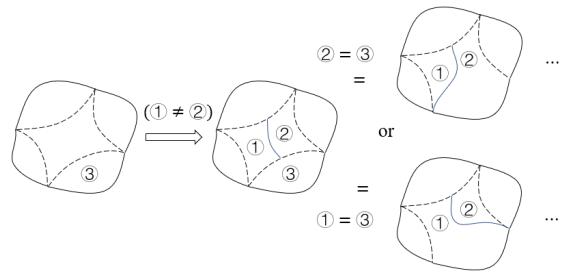
The difficulty of solving the coloring problem is that, although the points are gathered into a same cell, they can be filled in with different colors, instead of only being seen as a whole and drawn with a single color, see an easy example in Figure 5. We observe that the structure of the graph reflects in the connectivity of the topological edges, which is proved having only finite possible situations in IV-A.

In this section, we prove the finiteness of the number of divisions and introduce the enumerative solver through the example of solving a single cell with all its adjacent cells having been colored. We show that the simple cells with less than 4 topological edges can be enumeratively solved, and there are finite number of manners to divide a complicated cell into several simple cells, hence any cell can be solved.

A. Finiteness of Divisions

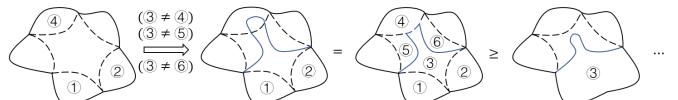
Since any path starting and ending at the boundary of a cell will divide the cell into two parts, there are infinite many physical solutions of dividing a cell into parts. However, there are only finite classes of them in the view of topological structure, because of the equivalence of physical divisions in the number of lift-offs.

See Figure 6(a), we show that the cutting paths which start or end at an ordinary point on some edges are unnecessary.

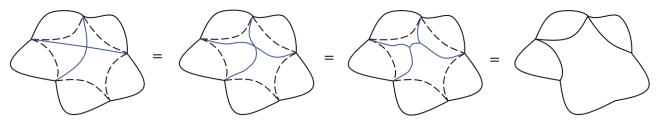


(a) The unnecessary of having a cutting path which starts or ends at an ordinary point of a topological edge.

Enforce the middle cell connecting ① and ②



(b) The unnecessary of having a cutting path going across an edge.



(c) The unnecessary of intersecting two cutting paths.

Fig. 6. Some physical divisions that are unnecessary or not optimal.

Let a cutting path ends at an ordinary point of the edge connecting cell 3. From the definition of a cutting path, it implicitly enforces cell 1 and cell 2 having different colors. Then we may assume $1 = 3$ or $2 = 3$ (when $1 \neq 3$ and $2 \neq 3$ the division is trivial), which, however, is equivalent to two other cutting paths that start at the endpoint of the edge. Do the same discussion on the other endpoint of this cutting path, we know that it is complete to only consider all cutting paths which start and end at the endpoint of some edges.

See Figure 6(b), we show that the cutting paths which go across some edges are unnecessary. Let a cutting path go across the edge that connects cell 4. The cutting path can be continuously transformed onto the edge without changing the cost. This division enforces the constraint of color that $3 \neq 4, 3 \neq 5, 3 \neq 6$. However, it prevents cell 5 and cell 6 from being colored together, because they are separated physically by the cell 3, which may increase the cost and is not optimal. So we can directly discard the cutting paths which go across edges.

See Figure 6(c), we show that the cutting paths need not go across each other. When two cutting paths intersect, we can change the belonging of the path segment, and then the cutting paths can be continuously transformed onto the existed topological edges. So it is complete to discard the choices of cutting paths which have intersections.

In conclusion, we only need to consider all cutting paths which start and end at the endpoint of some topological edges and do not go across each other. Hence, the total number of topological divisions is finite and we just need to go through all possible divisions.

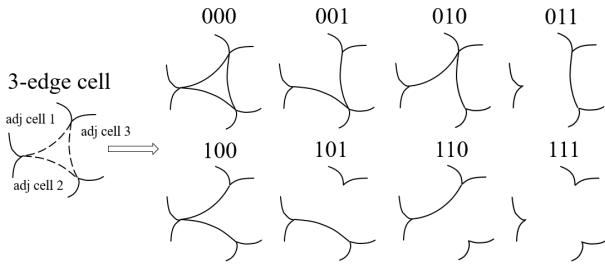


Fig. 7. All possible divisions of a 3-edge cell, which is the most complicated situation that need not further divisions. Although some of them are the same in the topological structure (e.g., 001, 010 and 100), or the division is impossible (e.g., we enforce 011 but the cell 1 and cell 2 do not have a same color), it has already been a finite problem, so we omit the description of further simplification.

B. Solution of Simple Cells

The following kinds of cells are so simple that can be solved directly without further divisions:

- (1) The cells containing less than four edges,
- (2) The cells with only one possible color,

because the cells of (1) cannot be divided further into several cells with less number of topological edges, and the cells of (2) have no other choice of color. We enumerate all possible topological divisions of a 3-edge cell (which is the most complicated case for direct enumeration) in Figure 7. We use a binary number to represent the connectivity of the edges, 1 for connection, 0 for disconnection and \times for the unspecified state. It is easy to see that there are at most 8 situations.

C. Solution of Complicated Cells

Following the idea of solving a simple cell, we use a binary number of length n to represent the connectivity of an n -edge cell, 1 for connection, 0 for disconnection and \times for the unspecified state. The continuous 1s imply that part of this cell must be painted with the same color as that of which the 1s specify. The 0 means that the topological edge between the cell and the corresponding adjacent cell is kept so that the colors must be different. The unspecified states \times imply the generation of a sub-cell. An example of solving a 4-edge cell is shown in Figure 8. Through specifying the position of 1s, there are less than $2^n \times m$ branches for an n -edge cell with m possible colors, so the problem is finite. The distinction between 0 and \times will be given in subsection IV-D.

D. Discussion of Creating Sub-Cells

When there exist such number lists in the connectivities:

$$1 \times \times 1, 1 \times \times \times 1, \dots$$

the cell is divided into parts whose colors are enforced to be different, so called sub-cells. See the case of $\times \times 11, \times 11 \times, 1 \times \times 1, 11 \times \times$ in Figure 8. The original cell becomes a new one with fewer edges, because some edges are replaced by a single edge. We use the bracket (\dots) in the binary number of the original cell to represent the generation of sub-cells. Do the same division for the sub-cells, any n -edge

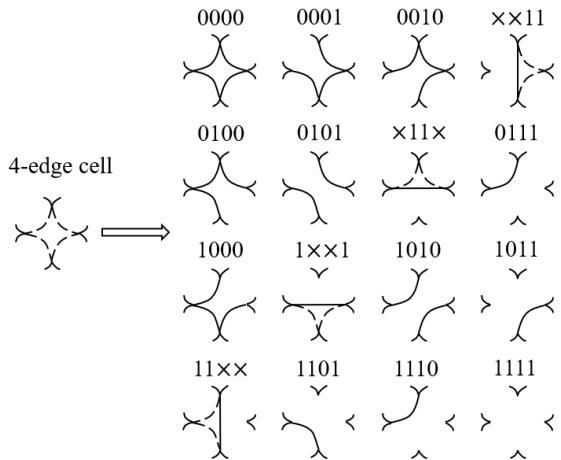


Fig. 8. The 2^4 possible divisions of an 4-edge cell. For the cell which has more than 3 edges, the sub-cell may be created. In this figure, the connectivities that correspond to generating a sub-cell is $\times \times 11, \times 11 \times, 1 \times \times 1, 11 \times \times$.

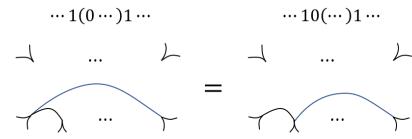


Fig. 9. The equivalence of moving the 0 outside the bracket. In order to reduce the number of edges of the sub-cell, we always enforce the sub-cell looking like the right side.

cell can be continuously divided into a set of 3-edge cells and then be solved enumeratively.

Since the sub-cell is generated from an original one, there are extra constraints on its connectivity specified by the previous division. However, these constraints cannot change this problem into one with polynomial time solution, so we just give some examples among them:

- (1) Single \times cannot form a sub-cell, because

$$\dots 1 \times 1 \dots = \dots 101 \dots \text{ or } \dots 111 \dots$$

but both conditions of the right side are considered in other branches. This is why there is no \times in Figure 7.

- (2) The 0s can be freely moved outside the bracket, because of the equivalence

$$\dots \times 1(0 \times \dots \times 1)1 \times \dots = \dots \times 10(\times \dots \times 1)1 \times \dots$$

and the same is true for the right bracket based on the symmetry of the number list, because the cutting graph makes sense only when the inner boundary two numbers are 1s. See Figure 9. This is why no brackets in Figure 8.

- (3) The new topological edge created by the cutting path must be kept (always 0, impossible to be 1 or \times) because it is manually created. Hence, for the entire problem, no extra possibilities appear after the divisions

V. ITERATIVE SOLVER

A. Iteration Process

Regarding the enumerative solver as a basic step, we iteratively solve the graph. Starting from a fully-unpainted

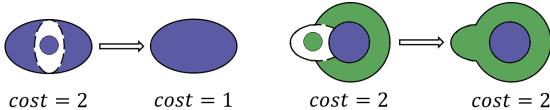


Fig. 10. Left: the middle cell connects two distinct cells, so the cost variation is $1 - 2 = -1$. Right: two edges connect with the same adjacent cell, then the cost variation is $1 - 1 = 0$ but not $1 - 2 = -1$.

graph, we choose an unsolved cell and enumeratively solve it. Assume that the cell has n -edges with m possible colors, following our previous discussion, there are at most $2^n \times m$ possible divisions. We create a branch for each possible solution of the closed cell, and in each branch the closed cell is filled in with the specified color (so it will not change any more). In the next iteration, we choose an unsolved branch, choose an unsolved cell in it and do the same steps as before. Note that the constraints given by the solved adjacent cells hugely restrict the possible solutions, because the state of an edge restricts the connectivity of the cells on both side. After iterative execution, all branches reach a contradiction or a valid coloring scheme. A valid coloring scheme for the graph uniquely specify the configuration to polish each point among its valid IK solutions. The algorithm runs like the deepest-first-searching (DFS) algorithm so that the memory requirement is not high. Since the execution is an exhaustive searching, all optimal physical cellular decompositions must be homeomorphic to one of our result schemes, with the position of the physical boundaries of the cells slightly different to the physical ones.

B. Calculation of Cost

The physical meaning of the cost for a (partly filled) graph is the number of pieces of color in the current graph. Describing the formula incrementally, after we solve a cell,

- (1) If its connectivity is all zero, then the cost will increase 1 after coloring this cell, because this cell forms a new piece.
- (2) If its connectivity has only one 1 connecting a solved cell, then the cost will not change, because this cell can be filled together with the connected adjacent cell.

(3) For its connectivity with i 1s, note that there may exist multiple edges which connect the same adjacent cell (See Figure 10). In order to be consistent with the physical meaning of the cost, if these edges connect j distinct solved cells, then the variation of cost is

$$\Delta\text{cost} = 1 - j$$

VI. EXPERIMENTAL RESULTS

Our algorithm works on any non-repetitive coverage task using non-redundant manipulators in any dimension. In this paper, experiments are taken by using a 5DOF manipulator to polish the surface of an object. And the manipulator should cover all reachable points even if it cannot fully cover the surface.

Before showing the experiments, we state the calculation of the least number of lift-offs under other cellular decompositions, from which we can see that the proposed algorithm outperforms other methods. In the simulated experiments, we

first show an object being polished at different poses, one casually and the other being designed precisely. We can see that the required number of lift-offs can distinguish them. Thus, the number can be seen as a criterion of the quality of the placement. Then, we show that our algorithm helps to get rid of bad configurations which result in extra lift-offs. In the real-world experiment, the physical coverage path is set to see the proposed algorithm running in reality. [Through the attached video, we can see that the manipulator finishes the coverage task under minimum number of lift-offs.](#)

Unless otherwise stated, the environment contains only the manipulator, the object and the ground plane. For easy talking, the collision models of everything are the same as their visualizations. And note that all figures in this section are just an example of the optimal solutions. As we have discussed before, there are infinite many physical cutting paths of decomposition that result in same number of lift-offs of the EE, and the choices of the coverage paths within each cell are also arbitrary.

A. Calculating least number of discontinuities for other methods

In above sections we have talked the proposed algorithm running on a simply-connected surface. As a simple corollary, if the surface has been divided by other cellular decomposition methods, assumed into n parts (where each cutting path implies a lift-off, then $n - 1$ lift-offs are required), we can apply the proposed algorithm in each cell to know the least number of lift-offs required in each single cell, assumed as p_i for the i -th cell. Finally, we will know that the least required number of lift-offs obeying the given cellular decomposition method is

$$n - 1 + \sum_{i=1}^n p_i$$

Obviously, applying the proposed algorithm to the cellular decomposition created by itself, the number n is optimal guaranteed by the exhaustive searching, and each cell can be covered without lift-offs, which means $p_i = 0, \forall i$. Hence, the proposed algorithm certainly outperforms other methods in the sense of the number of lift-offs.

B. Covering a hemisphere

In this experiment, through the polishing task of a semi-ellipsoid shape object, we show that the number of lift-offs provided by our algorithm can evaluate the quality of the placement of the object (or the manipulator).

See Figure 11, a common scene is that, we place the object casually (flatly at $(0.7m, 0m, 0m)$ related to the manipulator) and start the CPP task, since there is no apparent criterion on the quality of the placement. However, the proposed algorithm shows that such a placement of the object requires at least 3 lift-offs, but still fails in full coverage (the farthest area is unreachable), which is equivalent to at least 4 lift-offs. Instead of the usual setting, see Figure 12, if we precisely design the pose of the object (obliquely at $(0.7m, 0.1m, 0.08m)$), then not only the least number of lift-offs decreases to 2, but the

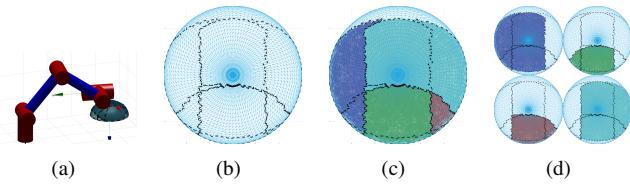


Fig. 11. (a) The object is casually placed. (b) The initial graph. (c) One optimal solution which requires 3 lift-offs, but the manipulator still cannot fully cover the farthest part of the mesh (the top area in the figure). (d) The reachable area of four kinds of valid configuration closed by the optimal solution in (c).

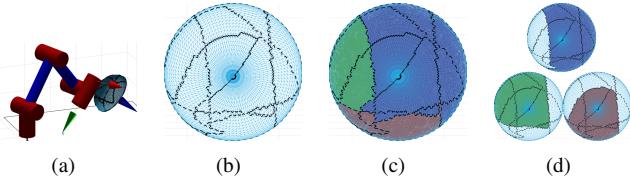


Fig. 12. (a) The object is placed obliquely. (b) The initial graph. (c) One optimal solution which only requires 2 lift-offs but realizes full coverage. (d) The reachable area of three kinds of valid configurations which are closed by the optimal solution in (c).

manipulator can fully cover the surface, which performs much better than the scene in Figure 11. Hence, the required number of lift-offs guides the user to try different poses of the object (or the manipulator) to have better performance on coverage.

C. Covering a pipe

In this subsection, through the polishing task of a half pipe, we show that the proposed algorithm can clarify the unnecessary configurations, avoid the “trapped” configurations which cause extra lift-offs.

The pipe is placed obliquely (at $(0.45m, 0.45m, 0.45m)$ related to the manipulator). Although the object is common, the normal of its surface varies for π rad, which causes difficulty for the manipulator. We show the initial graph and directly give its optimal solution in Figure 13. The optimal solution of this coverage task requires only 1 lift-off.

See Figure 14 for an explanation of the distinguishability of the useless configurations. The first kind of configurations and the third one are finally chosen by the optimal solution. It is worth saying that, although the second kind of configurations can polish large area without lift-offs, which looks appealing and is very likely to be chosen if the IK solutions are chosen randomly, it cannot reach the corners of the mesh (which are eventually covered by the other two ones). If the manipulator uses any of the configurations belonging to the second kind, after the coverage of the main part, sooner or later it has to waste two lift-offs to finish the full coverage, which leads to non-optimality.

D. Real World Experiments with Presence of Obstacles

In this subsection, we use a manipulator polishing the outer surface of a wok to show a physical coverage path generated based on the proposed cellular decomposition method. The physical coverage path uses simple back and force motions, whose generation is not part of our concern. The concatenation

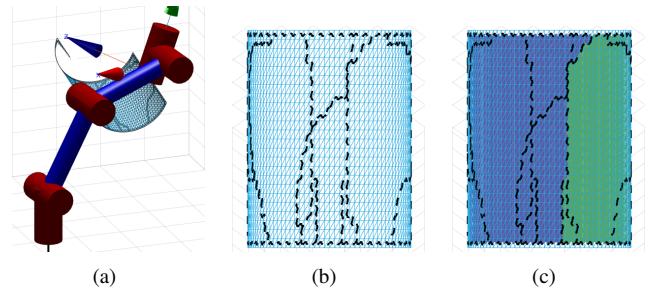


Fig. 13. (a) The manipulator is polishing the top-right corner where can only be polished through kinds of configurations. (b) The initial topological graph of this problem. (c) One optimal solution which requires 1 lift-off.

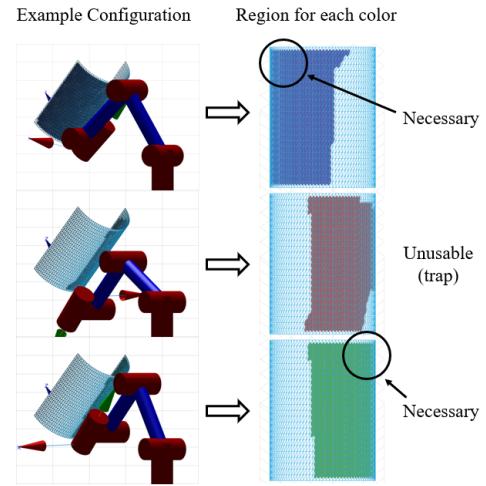


Fig. 14. Left: Example of three different kinds of configurations polishing a same point. Right: The coverage region for each kind of configurations. We omit the visualization of other kinds of configurations. Since there is only one color to cover the top corners (the first one for the top-left corner and the third one for the top-right corner), once any configuration belonging to the second kind is mistakenly chosen to cover any area of the surface, sooner or later we have to change to the first one and the third one to finish the full coverage of the surface, which wastes an unnecessary lift-off.

between paths in different cells are created by demonstration. The manipulator is UR5, with its last joint abandoned, equivalent to 5DOF, which is non-redundant. Since the hybrid position/force control is beyond our contribution, we do not involve the real contact.

See Figure 15, the nearest and farthest part of the wok are unreachable. Considering the two points shown in Figure 15(d)(e), the manipulator has to keep its wrist “above” its forearm in order to avoid collision between them, which leads to the requirement of both shoulder-left configurations and shoulder-right configurations. The total number of lift-off is 1. Since each color covers some points which only have one possible color, the solution is definitely optimal. Note that any division keeping the connectivity is optimal, so we may just divide the surface through the middle.

See Figure 16, the manipulator is obstructed by a cylindrical obstacle. Since the obstacle may collide with the upper-arm, fore-arm or the EE, and the wrist may collide with the fore-arm, to avoid all collisions, the least number of lift-offs is 2. Similarly, each color covers some points which only have one

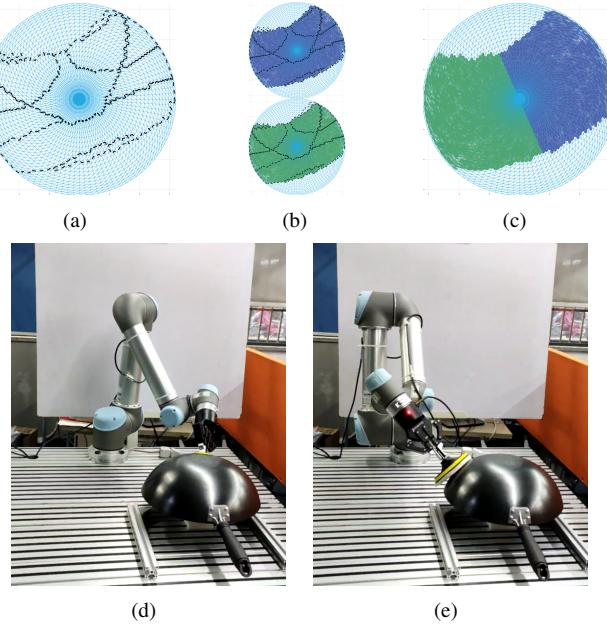


Fig. 15. (a) Initial graph. (b) The reachable area of two kinds of configurations which are closed by the solution in (c). (c) The cutting graph is arbitrary, so we may divide the graph through the middle. (d)(e) Example of the extreme poses of the two kinds of configurations.

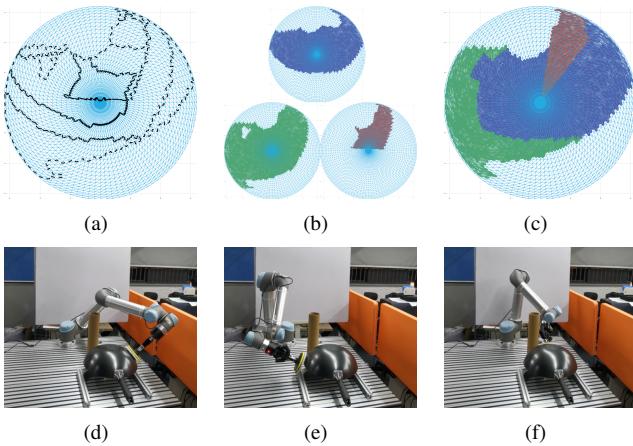


Fig. 16. (a) The initial graph. (b) The reachable area of three kinds of configurations which are closed by the solution in (c). (c) One optimal solution. (d)(e)(f) Example of the three kinds of configurations, where (d) The shoulder-left configurations avoid collision between the upper-arm and the obstacle. (e) The wrist is above the fore-arm so that the wrist will not hit the fore-arm. (f) The only valid configuration to cover this point is putting the wrist below the fore-arm.

possible color, so the solution is also optimal.

VII. CONCLUSION

In this paper, we first prove that the least number of the discontinuities is independent to the choice of the coverage path, thus becomes a criterion evaluating the quality of the placement of the manipulator (or the object), which may contribute to the mobile manipulator or the designing of the assembly line. Then, the proposed algorithm provides a novel cellular decomposition strategy, which, after applying the conventional CPP algorithm in each cell, generates the

result coverage path containing the least number of discontinuities, which is verified through simulated and real-world experiments. Also, as a direct corollary, applied to the result of other cellular decomposition methods, the proposed algorithm can tell the user the least number of discontinuities obeying the given cellular decomposition.

Because of the transition strategy which is more complicated than usual movement for coverage required by the discontinuities, and due to the unreachability of the optimal coverage path, an NP problem, the optimality criterion of the coverage path that ensuring least number of discontinuities becomes more significant, and reducing the number of discontinuities is practical to reduce the cost of the coverage task, which is solvable through the proposed algorithm.

In the future, a more complicated situation will be considered, where the initial topological graph contains ring-like cells, which can be broken during the iterative solving process. And the real contact process will be evolved to quantify the energy saving.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

- [1] G. Oriolo and C. Mongillo, "Motion planning for mobile manipulators along given end-effector paths," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 2005.
- [2] C. C. Cheah, S. Kawamura, and S. Arimoto, "Brief stability of hybrid position and force control for robotic manipulator with kinematics and dynamics uncertainties," *Automatica*, vol. 39, no. 5, pp. 847–855, 2003.
- [3] D. Heck, A. Saccon, N. Wouw, and H. Nijmeijer, "Switched position-force tracking control of a manipulator interacting with a stiff environment," *Proceedings of the American Control Conference*, vol. 2015, pp. 4832–4837, 07 2015.
- [4] S. S. S. Mirrazavi and B. Aude, "A dynamical-system-based approach for controlling robotic manipulators during noncontact/contact transitions," *IEEE Robotics & Automation Letters*, vol. 3, no. 4, pp. 2738–2745.
- [5] J. Porta and L. Jaillet, "Path planning on manifolds using randomized higher-dimensional continuation," vol. 68, pp. 337–353, 01 2010.
- [6] J. M. Porta, L. Jaillet, and O. Bohigas, "Randomized path planning on manifolds based on higher-dimensional continuation," *International Journal of Robotics Research*, vol. 31, no. 2, pp. 201–215, 2012.
- [7] P. Atkar, H. Choset, and A. Rizzi, "Towards optimal coverage of 2-dimensional surfaces embedded in \mathbb{R}^3 : choice of start curve," in *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '03)*, vol. 4, pp. 3581 – 3587, October 2003.
- [8] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms," in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 1, pp. 27–32 vol.1, May 2001.
- [9] J. E. Solanes, L. Gracia, P. Munozbenavent, A. Esparza, J. V. Miro, and J. Tornero, "Adaptive robust control and admittance control for contact-driven robotic surface conditioning," *Robotics and Computer-integrated Manufacturing*, vol. 54, pp. 115–132, 2018.
- [10] J. E. Solanes, L. Gracia, P. Munozbenavent, J. V. Miro, C. Perezvidal, and J. Tornero, "Robust hybrid position-force control for robotic surface polishing," *Journal of Manufacturing Science and Engineering-transactions of The Asme*, vol. 141, no. 1, p. 011013, 2019.
- [11] F. Paus, P. Kaiser, N. Vahrenkamp, and T. Asfour, "A combined approach for robot placement and coverage path planning for mobile manipulation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.
- [12] H. Choset, "Coverage for robotics – a survey of recent results," *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1, pp. 113–126, 2001.

- [13] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [14] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 4, pp. 462–472, 1990.
- [15] H. Choset, K. M. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementation*. MIT Press, 2005.
- [16] H. Choset and P. Pignon, "Coverage path planning: The boustrophedon cellular decomposition," pp. 203–209, 1998.
- [17] H. Choset, "Coverage of known spaces: The boustrophedon cellular decomposition," *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000.
- [18] H. Choset, E. U. Acar, A. A. Rizzi, and J. Luntz, "Exact cellular decompositions in terms of critical points of morse functions," vol. 3, pp. 2270–2277, 2000.
- [19] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks," *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.
- [20] P. A. Jimenez, B. Shirinzadeh, A. E. Nicholson, and G. Alici, "Optimal area covering using genetic algorithms," *international conference on advanced intelligent mechatronics*, pp. 1–5, 2007.
- [21] R. Mannadiar and I. Rekleitis, "Optimal coverage of a known arbitrary environment," pp. 5525–5530, 2010.



Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.